



Background

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

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

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

Hardware Requirements

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

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



Software Requirements

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

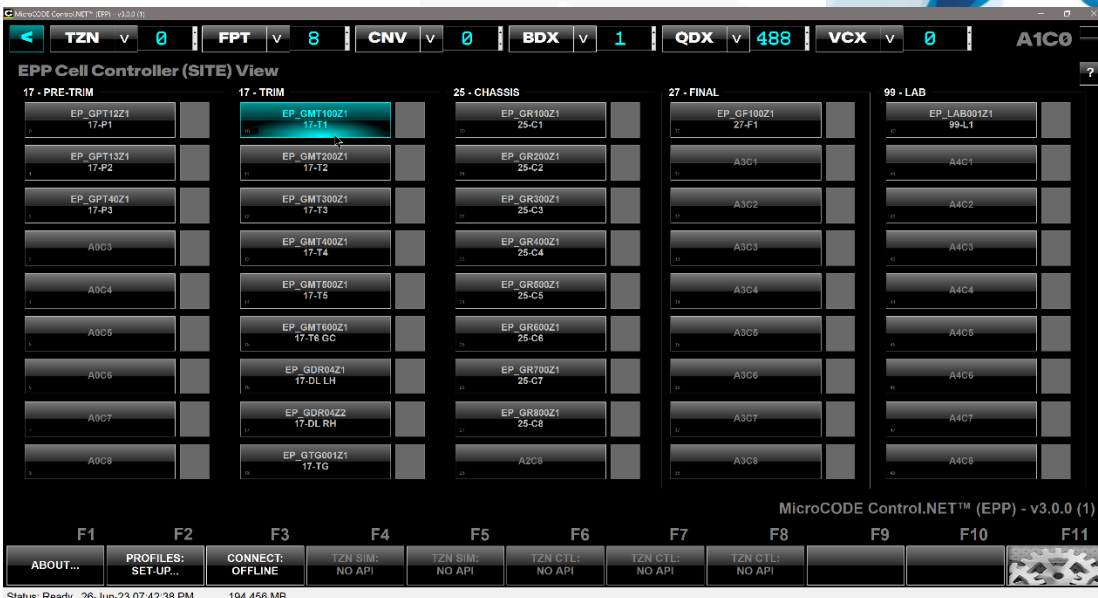
- **Windows 10** – installed on the PC
- **Microsoft .NET 4.5 or higher** – installed on the PC
- MicroCODE **Control** Windows 10 .NET Application
Note: .NET Framework 4. is always available on a Windows 10 PC*
- MicroCODE Application Interface PLC (**API**) program imported into the target EPP Cell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

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



4.0





Control App Software

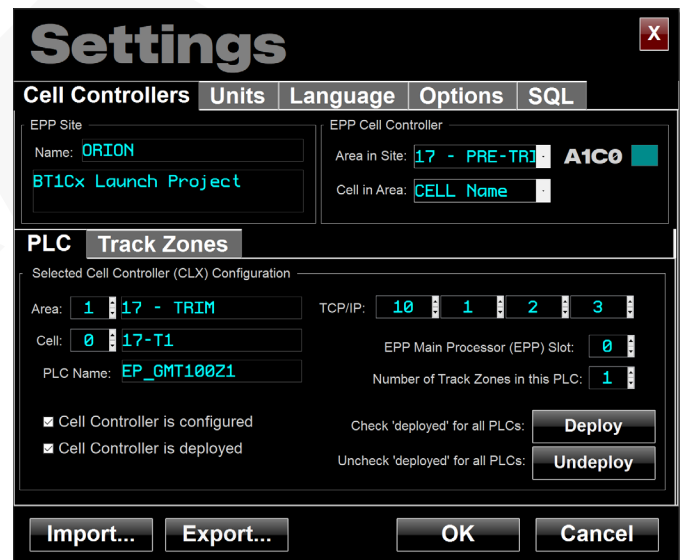
Application Framework

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 EPP SQL Database. **New in v1.0**
- **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 EPP** App provides the following EPP control features:

- **Conveyor Simulator Control** – supports direct control of the Conveyor Simulation built into the EPP 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 EPP 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.s.
- **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 EPP PLC, i.e.: The Line Tracking Image in EPP or the GEPICS Buffer in EPP.
- **GSIP Defect Queue Viewer** – a built-in tool to examine the Vehicle Defect Queue in the EPP 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 EPP PLC; i.e.: you can watch Trace Data being collected as you perform SATs*.
- **Requires no change to existing EPP 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 EPP v1.0.0 (Lake Orion)** – the interface program (**prgAPI**) isolates the Windows App from differences within the EPP PLC programs. But internal EPP changes between v1.0.0 and v1.4.x.x forces MicroCODE to deprecate support for the pilot version of EPP, with GM’s approval.

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

EPP Dependencies

The **MicroCODE Control for EPP** App requires the following items to remain unchanged within the EPP Processor (**EPP**), where the **n** represents the Track Zone # suffix or Conveyor specific name:

- **Configuration** – the “**ConfigControl_n**” and “**ConfigArraySize_n**” Controller Level Tags, their User-Defined Datatype (UDT) and all sub-Datatypes must remain in their EPP v1.0.0 (Lake Orion) form. Additions may be made, but all existing data items and their definitions must remain unchanged.
- **Conveyor Subsystem** – the “**GCSSTZ1**” Conveyor Program naming convention must be followed.
- **Line Tracking Subsystem** – the “**LineTrackingID_n**” and the “**TrackingImage_n**” Controller Level Tags, their User-Defined Datatype (UDT) and all sub-Datatypes their EPP v1.0.0 (Lake Orion) form. All existing data items and their definitions must remain unchanged. The maximum Build Data Packet size supported is the EPP Tracking is 1,000 total bytes.
- **Vehicle Quality Subsystem** – the Defects Queue “**QDIQueue**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their EPP v1.0.0 (Lake Orion) form. All existing data items and their definitions must remain unchanged.
- **Vehicle Quality Subsystem** – the Trace Queue “**TRaceQueue**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their EPP v1.0.0 (Lake Orion) form. All existing data items and their definitions must remain unchanged.
- **Build Data Subsystem** – the “**GEPICS_nODD1App**” Controller Level Tags, their User-Defined Datatype (UDT) and all sub-Datatypes must remain in their EPP v1.0.0 (Lake Orion) 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.
- EPP “**ErrorProofing**” SQL Database must remain as defined in the EPP v1.0.0 (Lake Orion) form.

The **MicroCODE Control for EPP** App requires the following guidelines be followed in the EPP PLC code:

- 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 EPP program 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 EPP functions.

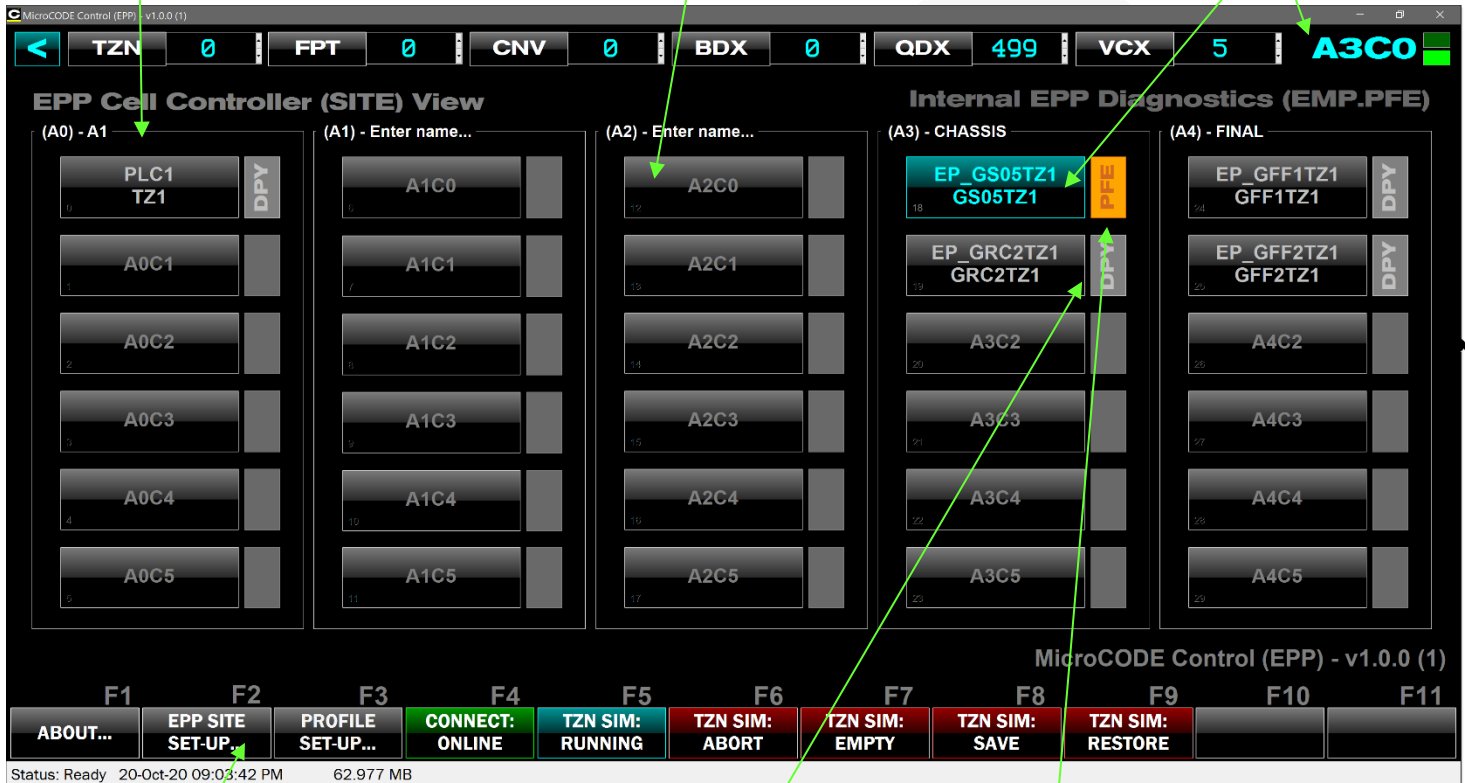
New in v1.0, this entire **SITE** layer was added to the Control App.

EPP SITE Summary Screen

EPP Area
An Area covered by up to (n) Cell Controllers. EPP supports up to (n) Areas. (30) Cells are shown on this screen by default.

EPP Cell Controller
A collection of up to (9) Track Zones, All controlled by a single PLC: EPP.

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 EPP SQL Database.

Cell Status: "DPY"
Indicates a Cell still needs to be deployed.

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



App Overview – dynamic visualization

The main App Screen is now free-form and re-designs itself based on the EPP SQL database layout.

The current limit is 10 Areas x 10 Cells each, but this can be changed in seconds.

New in v3.0, this entire **SITE** layer conforms to the EPP SQL Database layout.

EPP SITE Summary Screen – dynamic presentation

The screenshot displays the EPP Cell Controller (SITE) View interface. At the top, there is a navigation bar with buttons for TZN, FPT, CNV, BDX, QDX, and VCX, each with a dropdown arrow and a '0' indicator. The main area is a grid of cells, each containing an identifier. The grid is organized into columns labeled 17 - PRE-TRIM, 17 - TRIM, 25 - CHASSIS, 27 - FINAL, 99 - LAB, A5, A6, A7, A8, and A9. The cells contain various alphanumeric codes, such as EP_GPT12Z1, EP_GMT100Z1, EP_GR100Z1, EP_GPT13Z1, EP_GMT200Z1, EP_GR200Z1, EP_GPT40Z1, EP_GMT300Z1, EP_GR300Z1, EP_GMT400Z1, EP_GR400Z1, EP_GMT500Z1, EP_GR500Z1, EP_GMT600Z1, EP_GR600Z1, EP_GDR04Z1, EP_GRT00Z1, EP_GDR04Z2, EP_GRT00Z1, EP_GTG00Z1, and EP_LAB001Z1. A status bar at the bottom shows 'Status: Ready 05-Jun-23 07:43:20 AM' and '148.876 MB'. The bottom of the screen features a row of function keys (F1-F11) with corresponding labels like 'ABOUT...', 'PROFILES: SET-UP...', 'CONNECT: OFFLINE', and 'TZN CTL: NO API'.





EPP CELL Control Screen

The screenshot shows the EPP CELL Control Screen with the following callouts:

- Track Zone Selection & Viewer**: Points to the 'TZN' button and the 'EPP Track Zone (TZN) View' section.
- Footprint / Job Selection & Viewer**: Points to the 'Live Conveyor Status for current Footprints' callout.
- Conveyor Selection & Viewer**: Points to the 'Live Job Status for current Footprints' callout.
- Build Data 'Seed' Selection & Viewer**: Points to the 'GRC1TZ1' button.
- GSIP Queue Defect Viewer**: Points to the 'GRC1TZ1' button.
- VCVS Queue Trace Viewer**: Points to the 'GRC1TZ1' button.
- Site Set-Up**: Points to the 'EPP SITE SET-UP...' button (F2).
- Job Control Profiles (JCP)**: Points to the 'PROFILE SET-UP...' button (F3).
- Connect**: Points to the 'CONNECT: ONLINE' button (F4).
- SIM: Start**: Points to the 'TZN SIM: RUNNING' button (F5).
- SIM: Abort**: Points to the 'TZN SIM: ABORT' button (F6).
- CNV: Simulator Controls**: Points to the 'CONNECT: ONLINE' button (F4).



ControlLogix Integration

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

Name	Value	Style	Data Type
API		{-}	udtAPI
API_ClearFault		{-}	udtAPI_FaultRecord
API_Enum08		{-} Decimal	SINT[256]
API_Enum16		{-} Decimal	INT[256]
API_Enum32		{-} Decimal	DINT[256]
API_EventData		{-}	udtAPI_EventData
API_Semaphore		{-}	udtAPI_Semaphores
API_SimResponse		{-}	udtAPI_SimResponse
API_SimState		{-}	udtAPI_SimState
API_State		{-}	udtAPI_States

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

API Program
Application Programming Interface for C# App.

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

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.

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

- uc_TraceQueueData
- uc_ZA_EP01toCBZA2AInterlocks
- udtAPI**
- udtAPI_CnvInputs
- udtAPI_CnvSimulation
- udtAPI_EppConfig
- udtAPI_Event
- udtAPI_EventData
- udtAPI_FaultRecord
- udtAPI_Flags
- udtAPI_Hal
- udtAPI_Inputs
- udtAPI_Job_CSJ
- udtAPI_Outputs
- udtAPI_Semaphores
- udtAPI_SimResponse
- udtAPI_SimState
- udtAPI_States
- udtAPI_TimeStamp
- udtAPI_TznCommands
- udtAPI_TznConfig
- udtAPI_TznConveyor
- udtAPI_Version





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



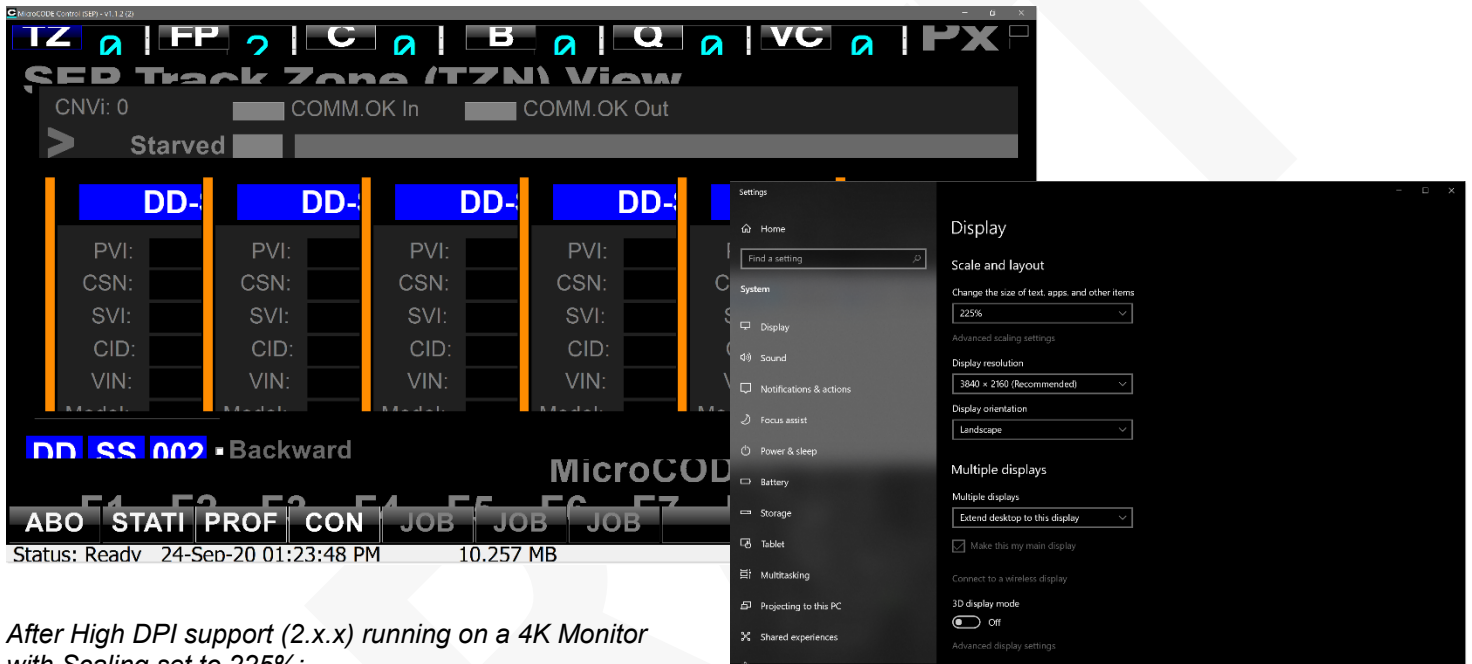
New in Version v1.0.0 a (0)

The following features were added to **MicroCODE Control (EPP)** 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 EPP Screen

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

These Cell Controllers are divided into five (5) Areas, each capable of hold six (6) Cell Controllers.

These numbers are arbitrary and can be changed.

This new screen can be manually configured from the EPP SITE SET-UP dialog box.

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

Each Panel's configuration holds:

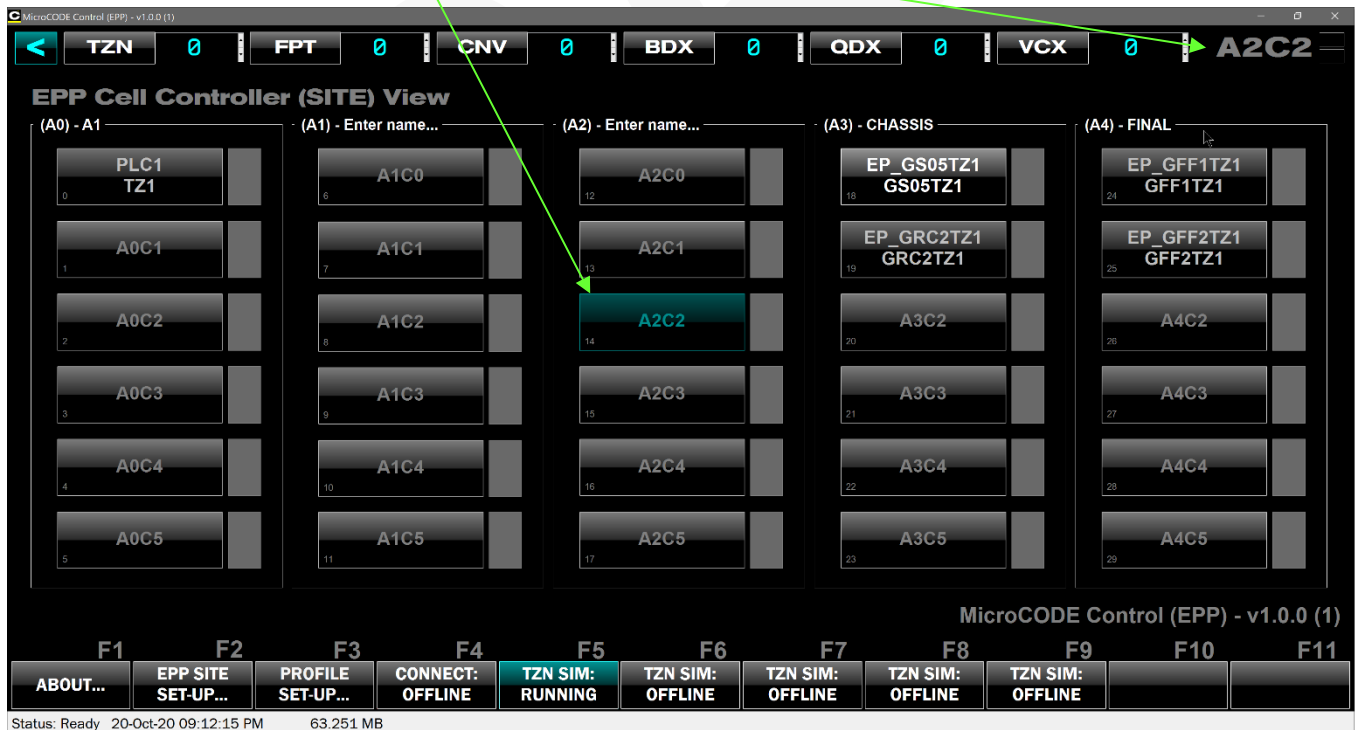
- Area Index
- User configured Area Name
- Configured Status
- Deployed Status
- Area Type: TRIM, CHASSIS, ENGINE, FINAL

Each Cell Controller's configuration holds:

- Area 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 v1.0, 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 Dialogbox

This allows all the EPP Cell Controllers in a Shop to be configured from a single dialog box. A Shop / EPP Server can handle up to thirty (30) Cell Controllers.

Cell Controllers
New in v1.0, users can now configure all possible Controllers in a Site once.

SQL Configuration
New in v1.0, users can now pull all required configuration from a GM Site's EPP SQL Database.

Cell Selection
New in v1.0, 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 v1.0, 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. See the Track Zones tab.

App Configuration Import
New in v1.0, 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 EPP SQL DB.

EPP SQL Server

New in v1.0, 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 v1.0, users can now pull all required configuration from a GM Site's EPP SQL Database.

SEP Site

Cell Controller | Units | Language | Options | **SQL**

This tab defines the connection and usage of the SEP Server SQL Database. It allows this App to pull all required configuration data from that database, rather than reenter it here, which is also supported.

SEP Server TCP/IP: 172 | 18 | 208 | 1

Local SQL DB Instance: (localdb)\MSSQLLocalDB Use Local DB

SEP SQL Database Name: SEPDB

SEP SQL User Name: sepdb_ro

SEP SQL Password: [Masked]

NOTE: The User is locked to the common READ-ONLY account.

NOTE: The Password is changed by GM IT on a regular basis and must be acquired from them.

CLEAR from this App Only: **Clear ALL...** REPLACE from SEP SQL DB Query: **Query ALL...**

Clear (or Replace from SQL) all configuration from this App, Cell Controllers IPs, Track Zones, Footprints: **Clear TZNs...** **Query TZNs...**

Clear (or Replace from SQL) all TRACK ZONES and FOOTPRINTS ONLY: **Clear TZNs...** **Query TZNs...**

Import... **OK** **Cancel**

Erase Track Zones

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

Get Track Zones

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

Erase All Configuration

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

Get All Configuration

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



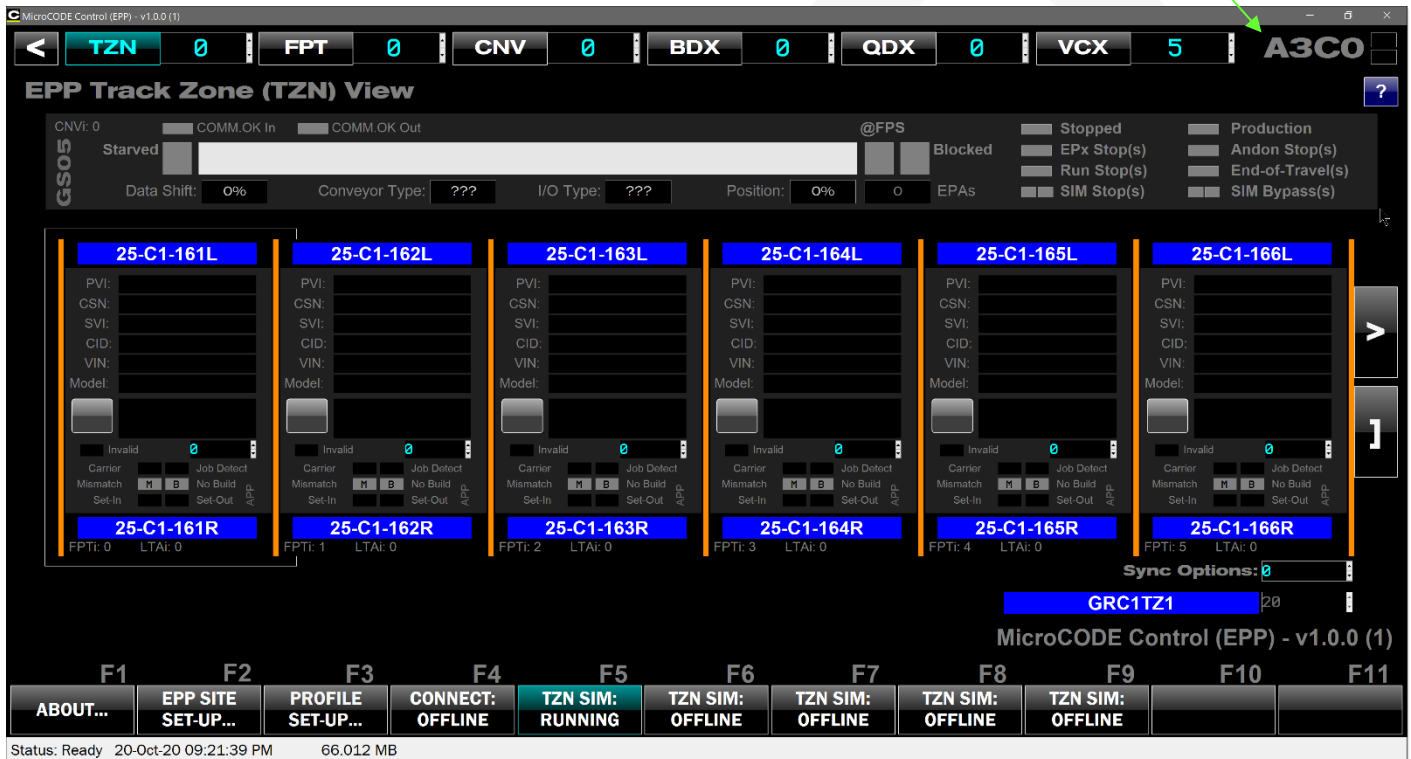


5) Automatic DSOA Placards

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 EPP 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 v1.0, the Active Cell Controller is always displayed in the upper right, this is the only Controller that can be written to by the App.





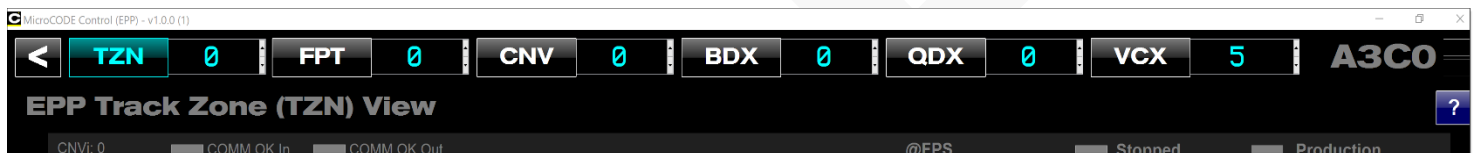
6) Track Zone Level Navigation

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 EPP Subsystems (PLC Programs) in both the EPP 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 along side the Cell selection buttons.

Cell Status: A3CO GS05TZ1 ? X

EPP Main Processor (EMP)

<div style="margin-bottom: 5px;">EMP EPP - Main Processor (EMP)</div> <div style="margin-bottom: 5px;">PFE Internal EPP Diagnostics (EMP.PFE)</div> <div style="margin-bottom: 5px;">CLX ControlLogix Hardware (EMP.CLX)</div> <div style="margin-bottom: 5px;">EPP Main EPP Program (EMP.EPP)</div> <div style="margin-bottom: 5px;">ODD GEP/CS Build Data Interface (EMP.ODD)</div> <div style="margin-bottom: 5px;">LTS Line Tracking Subsystem (EMP.LTS)</div> <div style="margin-bottom: 5px;">SCN Scanner Interfaces (EMP.SCN)</div> <div style="margin-bottom: 5px;">EPX Error Proofing Tasks (EMP.EPX)</div>	<div style="margin-bottom: 5px;">SIM Simulation Interface <Apps + Web> (EMP.SIM)</div> <div style="margin-bottom: 5px;">HMI PanelView Interface (EMP.HMI)</div> <div style="margin-bottom: 5px;">PMX Plant Monitoring Interface <GPM&C> (EMP.PMX)</div> <div style="margin-bottom: 5px;">UIX User Interface <Configuration> (EMP.UIX)</div> <div style="margin-bottom: 5px;">IOX External I/O Interface(s) (EMP.IOX)</div> <div style="margin-bottom: 5px;">CNV Conveyor Interfaces (EMP.CNV)</div> <div style="margin-bottom: 5px;">QAX Quality Andon Interface <QAS 1,2,4> (EMP.QAX)</div> <div style="margin-bottom: 5px;">VQX VQX Vehicle Quality Interface <GSIP> (EMP.VQX)</div>
---	--

12	46	20%
Controller Scan Time <Milliseconds> (EMP.CLX.CLS)	Controller I/O Connections (EMP.CLX.CLC)	Communication Time slice Remaining (EMP.CLX.TSR)

Key X

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

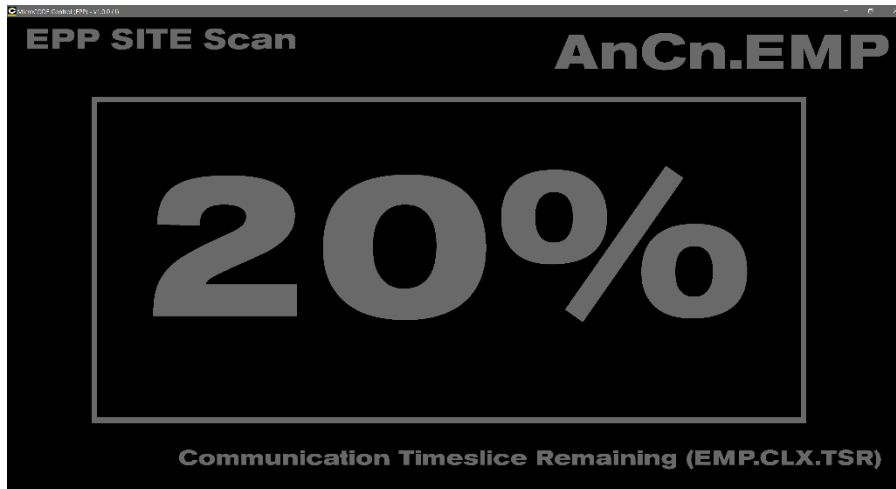




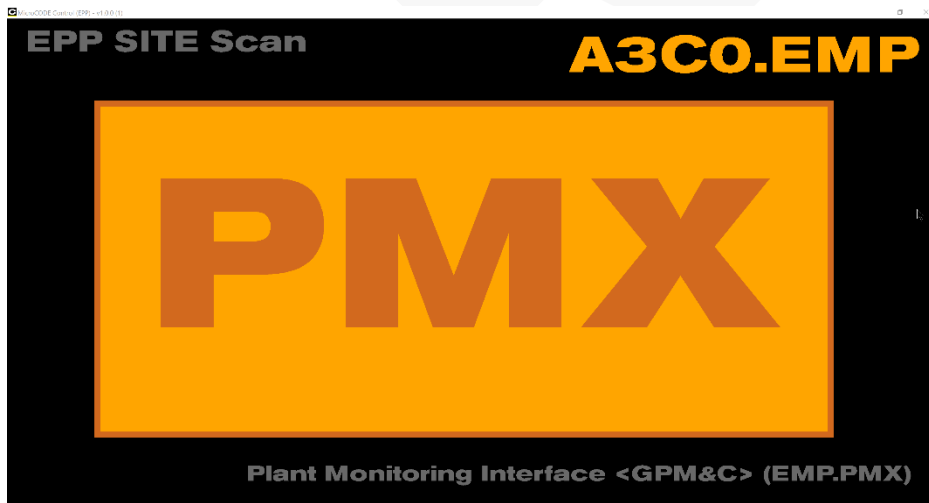
8) EPP Site Scan Display

A new display showing the health of the EPP Subsystems (PLC Programs) in both the EPP 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 EPP 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

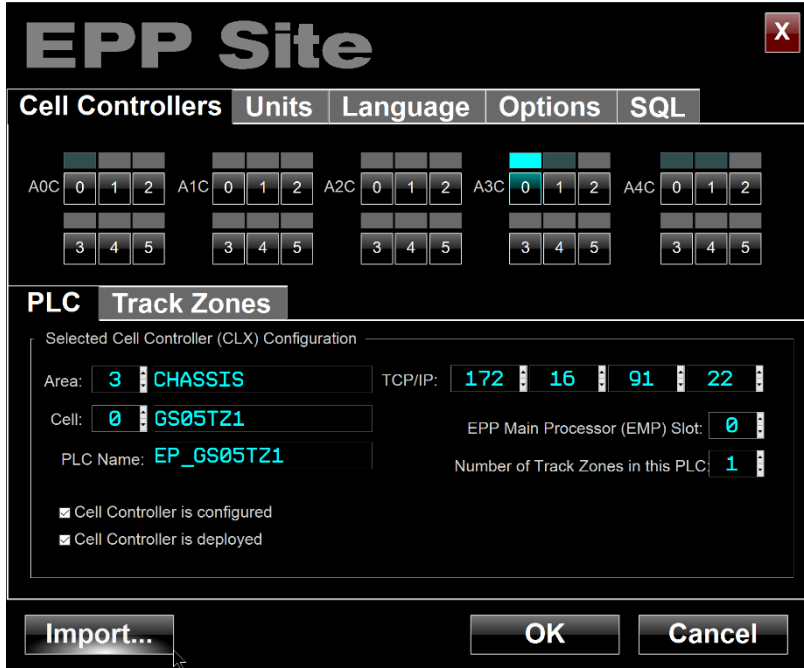
Our ControlLogix® 5580 controllers provide increased performance, capacity, productivity, and security to help meet the growing demands of smart machines and equipment for manufacturing. All ControlLogix 5580 controllers use the Studio 5000® design environment as the standard framework that optimizes 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.





11) SITE Configuration IMPORT

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





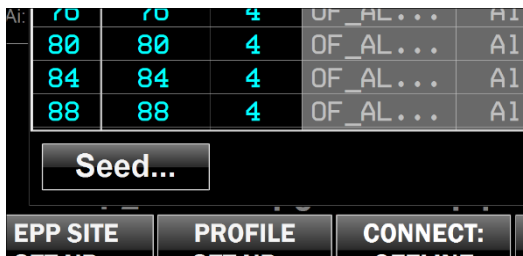
New in Version v1.0.0 a (2)

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

1) The 'Import...' feature of the GEPICS Build Viewer has been deprecated.

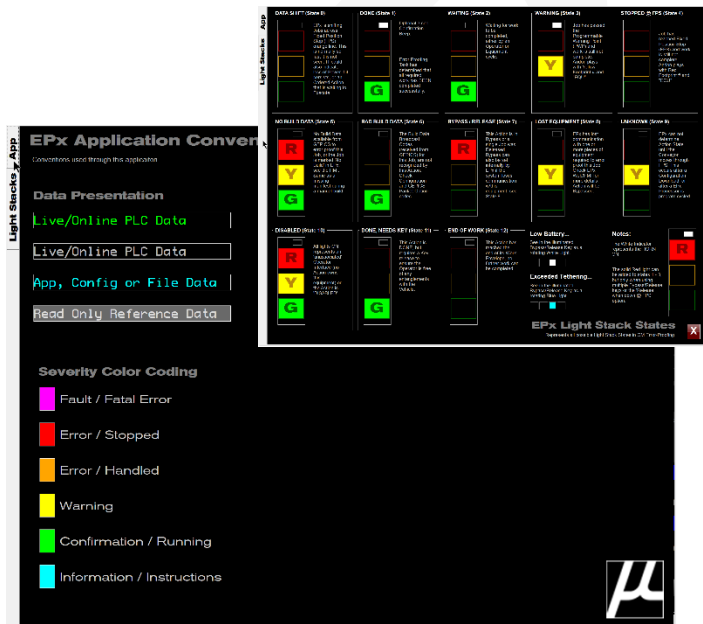
Because EPP requires two (2) Formats for every Track Zone (AREA and ZONE), you cannot simply change it on-the-fly as you can in the Control (EPP) App.

Instead of importing a new Format you must go back to the SITE: TRACK ZONE tab and configure the AREA and ZONE GEPICS Formats for your Track Zones.



2) 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.



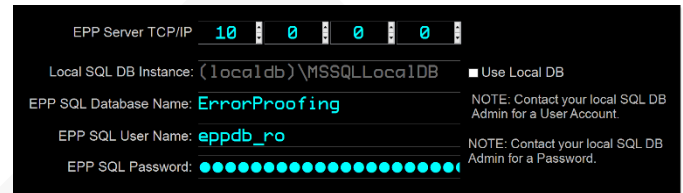
Corrected in Version v1.0.0 a (2)

The following defects were fixed in **MicroCODE Control (EPP)** 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.

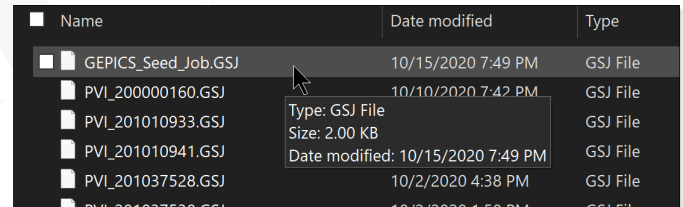


2) Cannot Import a GEPICS Seed Job

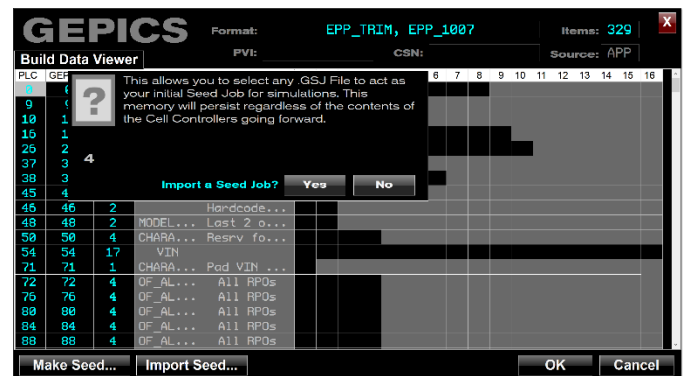
Issue: When the local PLC has no Build Data the U.I. does not have a feature to import a GEPICS Seed Job (.GSJ) from a file.

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

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



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





3) After selecting a GEPICS AREA or ZONE Format the TRACK ZONE configuration is not updated

Issue: In the SITE Dialog, while configuring AREA and ZONE Formats the dialog box is not refreshing after making a Format selection. Moving off the cell shows that the selection was actually made successfully.

PLC	Track Zones					
TZNi	Name	Footprints	FPTsi	Conveyor	AREA Format	ZONE Format
0	GRC1TZ1	7	0	GRC1_ALS		Y
1		0	7			N
2		0	7			N
3		0	7			N
4		0	7			N
5		0	7			N

Work-around: Just move to any other cell and come back.

Correction: The dialog was corrected to refresh immediately after a selection for AREA or ZONE.

PLC	Track Zones					
TZNi	Name	Footprints	FPTsi	Conveyor	AREA Format	ZONE Format
0	GRC1TZ1	7	0	GRC1_ALS	EPP_TRAIN.txt	Y
1		0	7			N
2		0	7			N
3		0	7			N

4) 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.

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

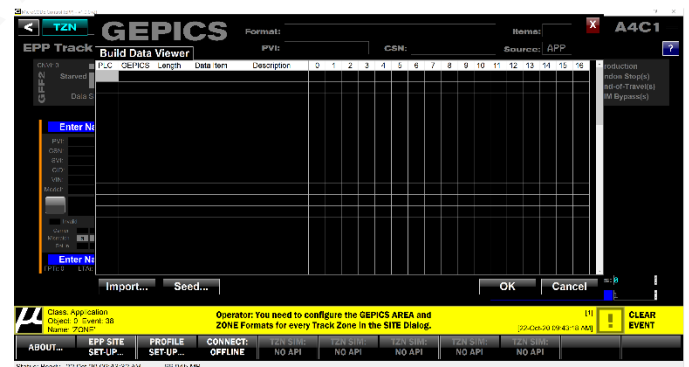
Issue: When opening the GEPICS Build Data Viewer if no GEPICS Format has been configured in the SITE: TRACK ZONE tab the App was displaying an unhandled exception. The user could continue working but it was annoying.



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	GFF2TZ1	0	0	GFF2_FPS		N
1		0	0			N
2		0	0			N
3		0	0			N

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





New in Version v1.0.1 a (1)

The following features were added to **MicroCODE Control (EPP)** 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.

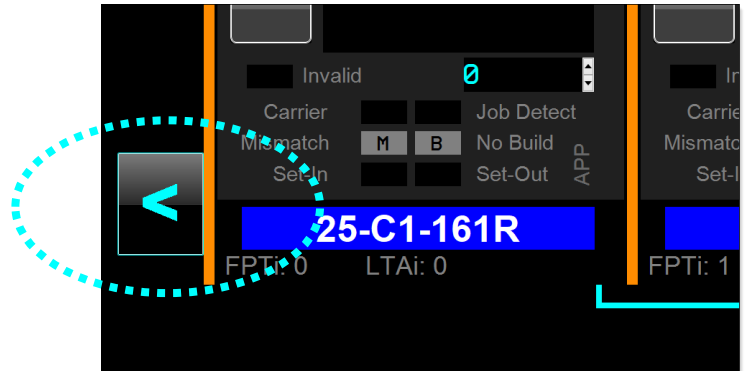
This is controlled by the new 'Job Focus' Left/Right Buttons in the Track Zone View.

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.

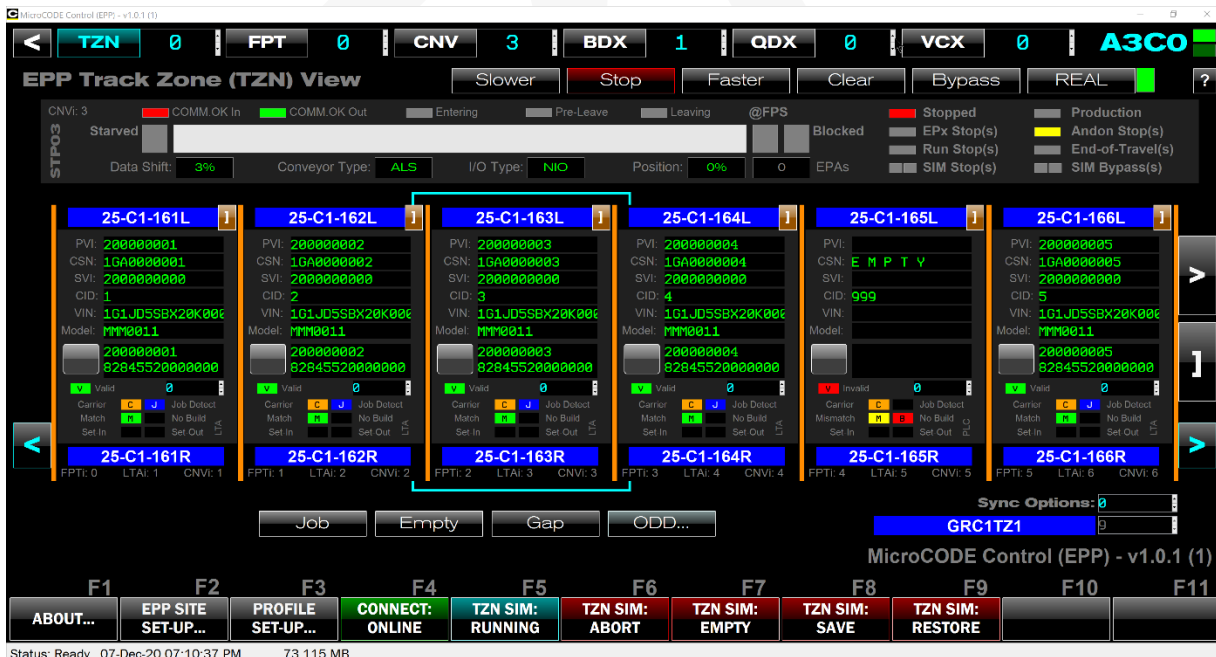
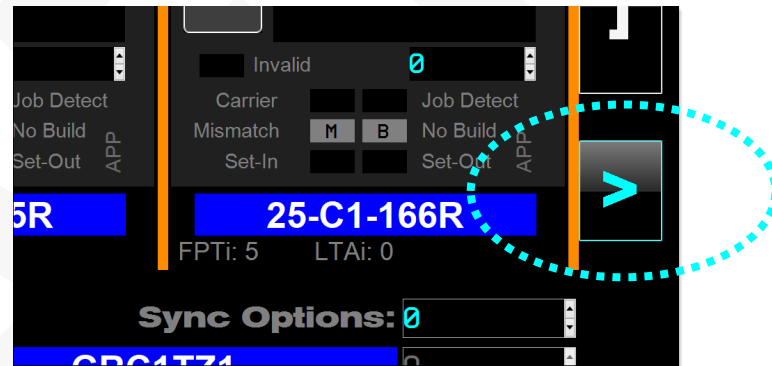
The **JOB**, **EMPTY**, and **GAP** commands follow the location of the 'Job Focus'.

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



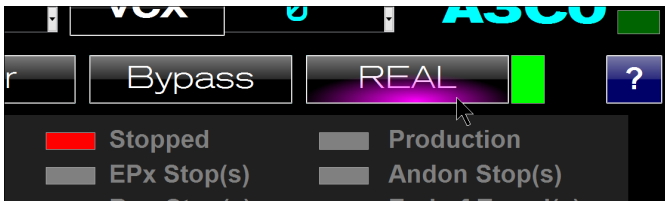


2) You can release the Conveyor from simulation but continue generating Jobs

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 v1.0.1 a (1)

The following defects were fixed in **MicroCODE Control (EPP)** 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 v1.0.1 a (2)

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

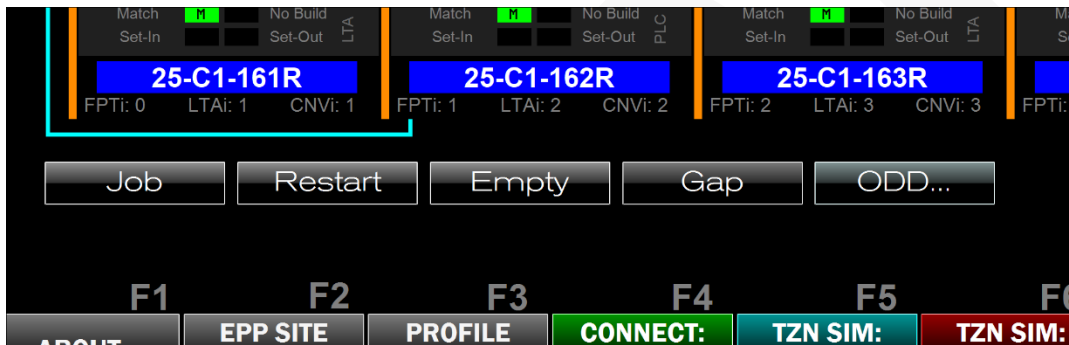
1) Complete SQL Integration

With the integration to the EPP 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 '.Valid' bit was set by the EPP 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 v1.0.1 a (2)

The following defects were fixed in **MicroCODE Control (EPP)** 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 v1.0.1 a (3)

The following features were added to **MicroCODE Control (EPP)** 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!”



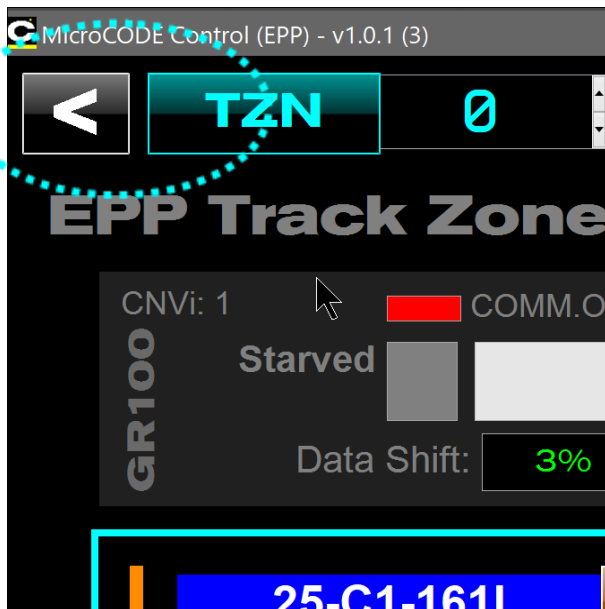


2) Improved 'Back' Function

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 v1.0.1 a (3)

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

1) Last Footprint name is missing

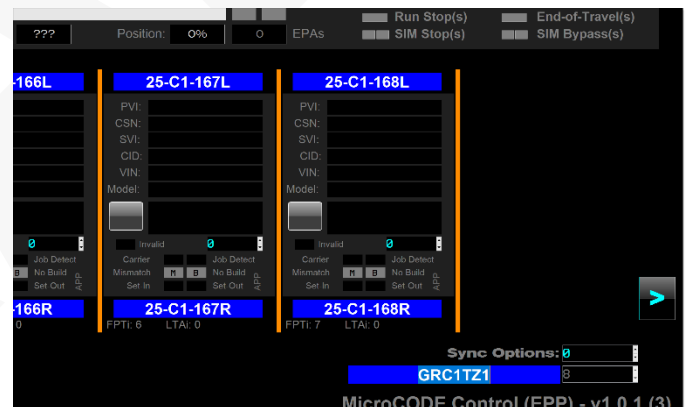
Issue: After pulling EPP configuration out of the EPP SQL Database the last Footprint names of each Track Zone.

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**.



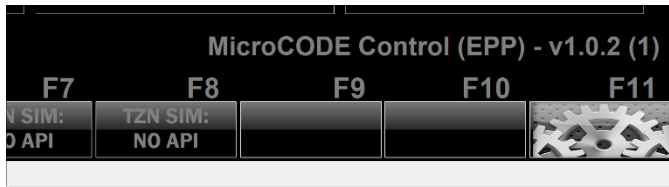


New in Version v1.0.2 a (1)

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

1) Standardizes SETTINGS dialog box in all MicroCODE Control Apps

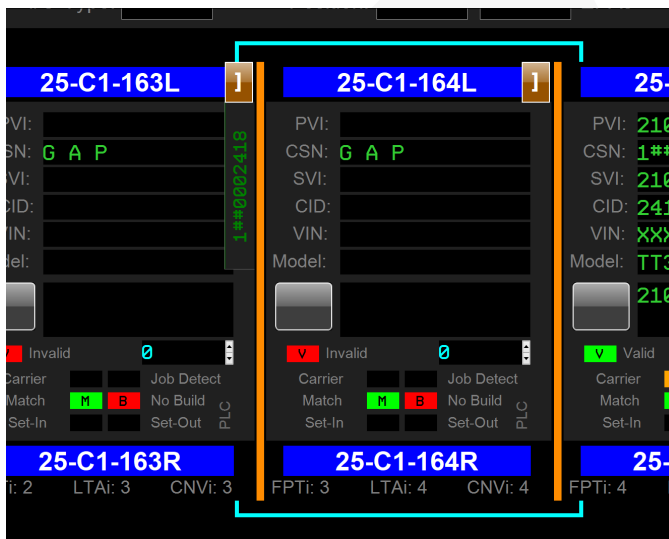
All Control Apps now have the 'SETTINGS' dialog boxes moved to the F11 function key and now show a typical 'Gear' image to aid in discoverability.



1) EPP 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 v1.0.2 a (1)

The following defects were fixed in **MicroCODE Control (EPP)** 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 v1.0.2 a (2)

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

None.

Corrected in Version v1.0.2 a (2)

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

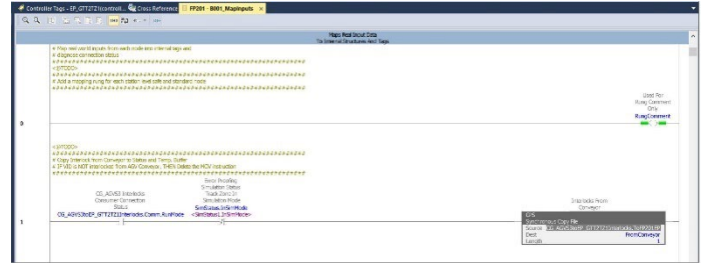
1) Added support for the new EPP PLC Conveyor Interface

Old EPP PLC Code:

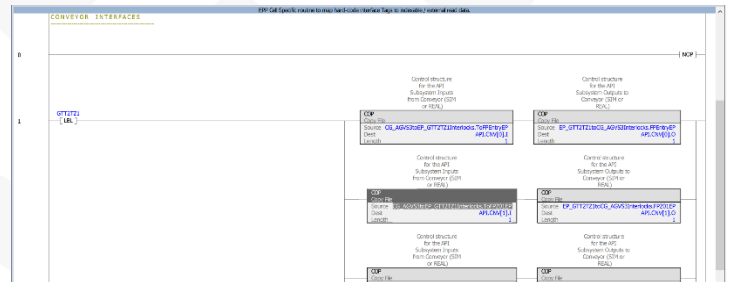
- Named Footprints in the Produce-Consume Tag from the Conveyor, e.g.: “FP201EP”

Name	Value	Data Type
CG_AGV53toEP_GTT2T21Interlocks		uc_CG_AGV53PLCtoEP_GTT2T21PLC
CG_AGV53toEP_GTT2T21Interlocks.Comm		CONNECTION_STATUS
CG_AGV53toEP_GTT2T21Interlocks.ToFFEntryEP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF200EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF201EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF202EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF203EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF204EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF205EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF206EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF207EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF208EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF209EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF210EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF211EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF212EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF213EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF214EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF215EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF216EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF217EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF218EP		zz_EPFFromConveyor
CG_AGV53toEP_GTT2T21Interlocks.ToFF219EP		zz_EPFFromConveyor

- CDE generated code referring to the ‘fixed’ Footprint Names... e.g.: “FP201EP”



- Control API following suit – “FP201EP” – note the MicroCODE App transferring into an intermediate indexed structure (API.CNV)

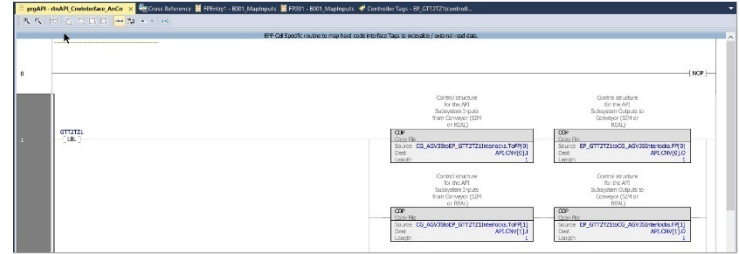




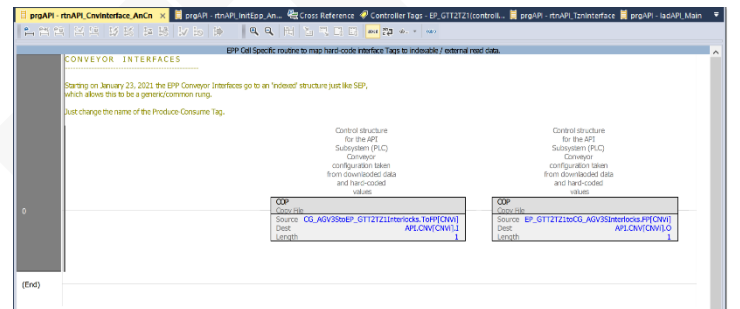
New EPP PLC Code:

- Generically indexed Footprints in the Produce-Consume Tag from the Conveyor, e.g.: "FP[2]" = FP201EP

Name	Value	Data Type
CG_AGV3StoEP_GTT2TZ1Interlocks	(-)	uc_CG_AGV3SPLCtoEP_GTT2TZ1PLC
CG_AGV3StoEP_GTT2TZ1Interlocks.Comm	(-)	CONNECTION_STATUS
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP	(-)	zz_EPFFromConveyor[21]
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[0]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[1]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[2]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[3]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[4]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[5]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[6]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[7]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[8]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[9]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[10]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[11]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[12]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[13]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[14]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[15]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[16]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[17]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[18]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[19]	(-)	zz_EPFFromConveyor
CG_AGV3StoEP_GTT2TZ1Interlocks.ToFP[20]	(-)	zz_EPFFromConveyor
CG_AGV3Controller	(-)	zc_ControllerRemWithStatus

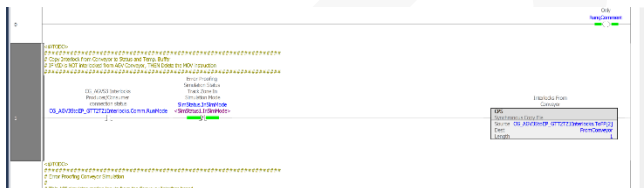


New MicroCODE Control App API was released along with v1.0.2 (2) to support this.



The routine `rtnAPI_CnvInterfac_AnCn` is now a single rung that references the overall Produced-Consumed Tag names for the Conveyor interface.

- CDE generated code now refers to the 'indexed Conveyor Interface Objects', just like SEP... e.g.: .FP[2] = FP201EP



Update your entire API:

- Delete the `prgAPI` program.
- Delete all API* Controller Level Tags.
- Delete all `udtAPI*` User Defined Datatypes.
- Import the new `prgAPI` supplied with this update.

- So now the App's API must be changed, and there is a mixture of EPP PLC versions in the field, i.e.: KUKA logic still uses the named Footprints.
- Notice that we anticipated this and tried to isolate my API from the actual construction of the EPP Conveyor interface by using an intermediate structure **API.CNV**.





New in Version v1.0.2 a (3)

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

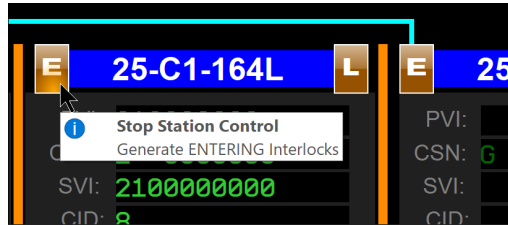
1) The App's Stop Station support now includes direct ENTERING/LEAVING Interlock support for Stop Stations

Requirement: Give Users comprehensive control over the simulation of Stop Stations.

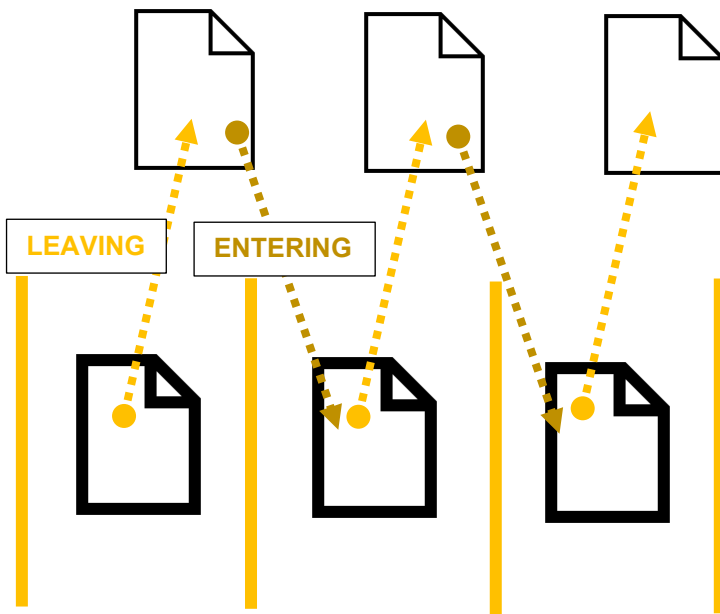
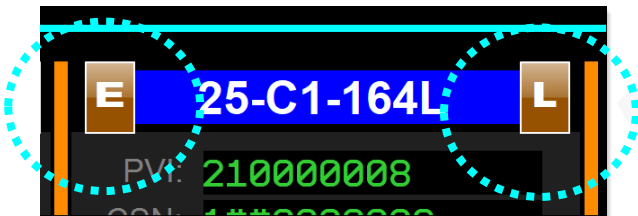
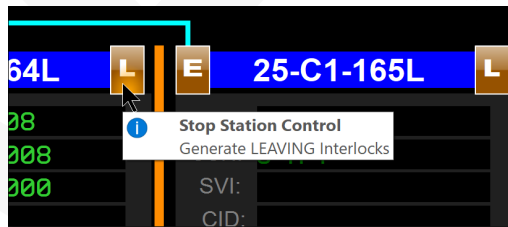
- Direct control of ENTERING Interlocks.
- Direct control of LEAVING Interlocks.
- Visualization of BUFFER Spaces.

Implementation: The App now displays new ENTERING and LEAVING control buttons on every Stop Station Footprint:

The ENTERING Button generates simulated Conveyor Interlocks to pull a JOB from the previous Footprint's BUFFER ("TrackingBufferN") into the selected Footprint's IMAGE ("TrackingImageN").



The LEAVING Button generates simulated Conveyor Interlocks to push a JOB out of the the selected Footprint's IMAGE ("TrackingImageN") into the selected Footprint's BUFFER ("TrackingBufferN").

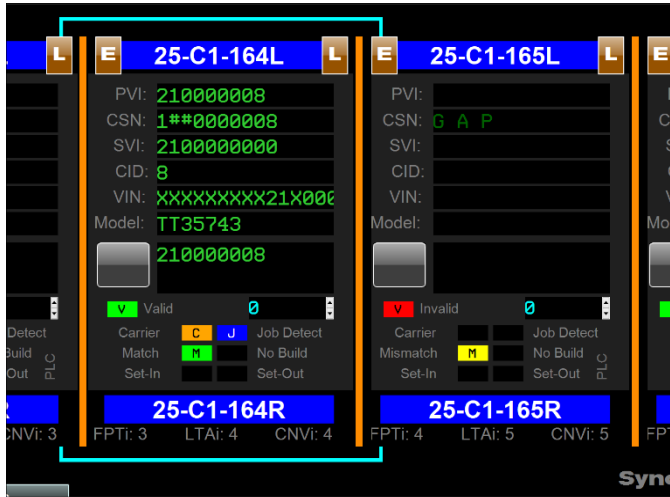


TrackingBufferN

TrackingImageN



Before LEAVING....



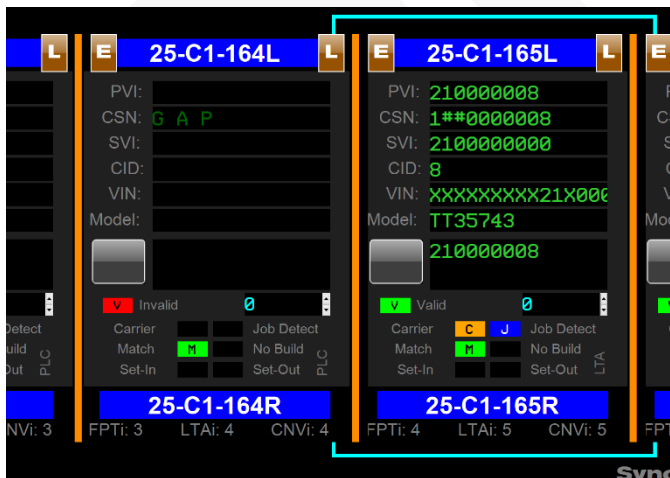
NOTE: There is a no JOB/CARRIER in the Tracking Buffer, indicated by the lack of a display on the right edge of the Footprint.

After LEAVING / Before ENTERING...



NOTE: There JOB has moved into the Tracking Buffer, indicated by the display of the CSN on the right edge of the Footprint, and the GAP left behind in the Footprint.

After ENTERING...



NOTE: The JOB has left the Tracking Buffer, indicated by the lack of a display on the right edge of the Footprint, and entered the next Footprint.





Corrected in Version v1.0.2 a (3)

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

No issue corrections.

DRAFT



New in Version v1.0.2 a (4)

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

1) Add support for new EPP AGV Conveyor Interface

Requirement: Give Users comprehensive control over the simulation of AGV Stop Stations.

- Direct control of ENTERING Interlocks.
- Direct control of LEAVING Interlocks.
- Visualization of BUFFER Spaces.

Implementation: This was an extension of the ALS Stop Station support. It required support for **Occupy** and **Empty** interlocks, as well as provided PVI and CARRIER ID like an AGV PLC.

Corrected in Version v1.0.2 a (4)

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

No issue corrections.





New in Version v1.0.2 a (5)

The following features were added to **MicroCODE Control (EPP)** 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 v1.0.2 a (5)

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

1) Corrected App interpretation of the EPP SQL DB relationships

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

Solution: The APP queries were corrected.

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

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

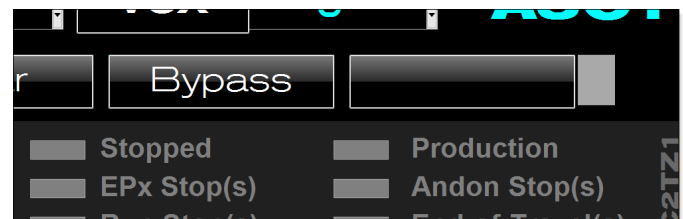
Solution: The API for the EPP PLCs required AREA and CELL be added to distinguish one from another.

The App only deals with one EPP 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 v1.0.2 a (6)

The following features were added to **MicroCODE Control (EPP)** 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 v1.0.2 a (6)

The following defects were fixed in **MicroCODE Control (EPP)** 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	MODE	20	7	NNNNNNNN	MODE	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	
PVI		0
CHARACTER_SPACES	Pad PVI to Length of 10	2
BSSEQNUM	Last 5 of BSSN	
CHARACTER_SPACES	Pad SVI to length of 16	
CSN		1





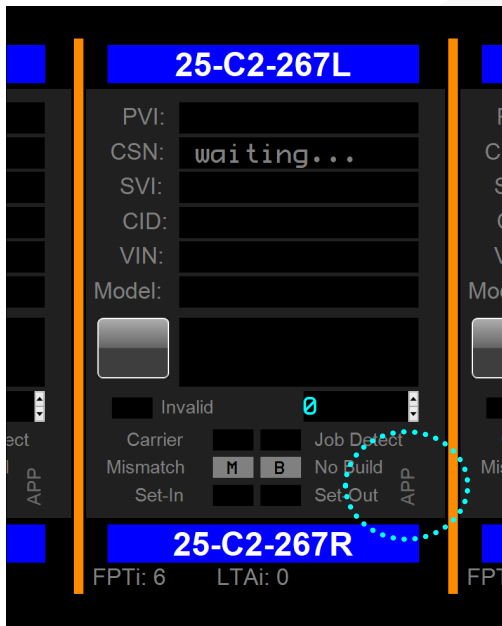
3) Additional Data Source Help Definitions

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

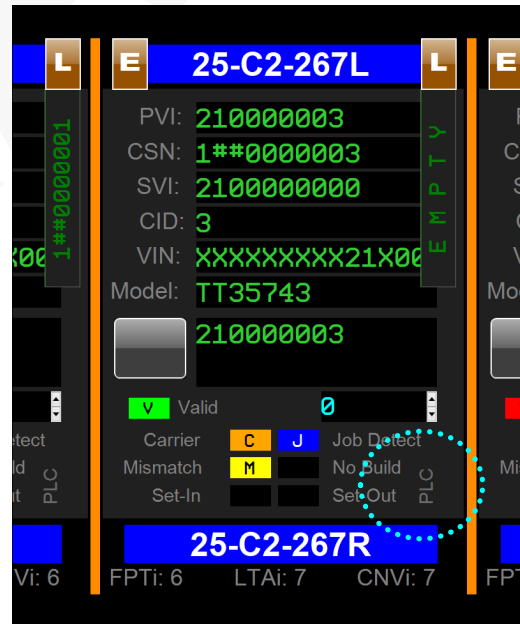
Implementation: Key App Objects now display the source of their data and the Help Screen was updated with the definitions of these sources.



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 EPP 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".





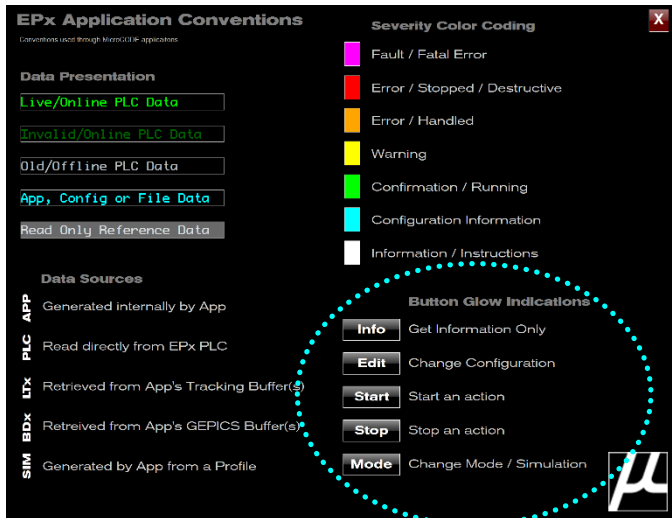
New in Version v1.0.2 a (7)

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

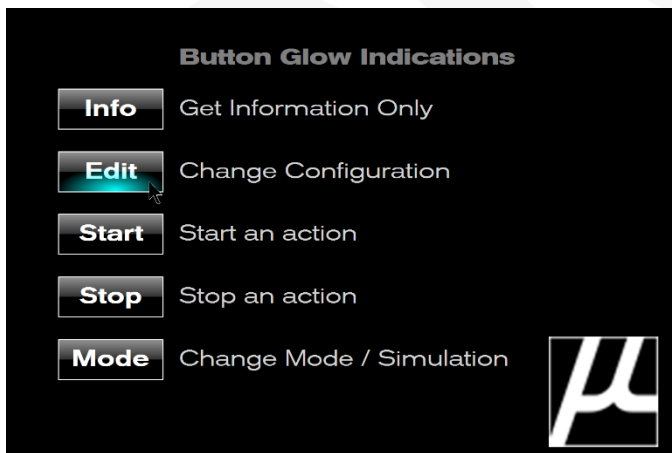
1) Help Screen – additional information

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.

+ LTS_LTA[0].BD.Byte[2754]	'\$24'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2755]	'\$01'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2756]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2757]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2758]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2759]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2760]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2761]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2762]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2763]	'\$00'	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2764]	'\$00'	ASCII	SINT
+ LTS_LTA[1]	{...}		udLTS_J
+ LTS_LTA[2]	{...}		udLTS_J

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	CIYX BB/...	2	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	X
1520	1594	8	E03 CIYX BB/...	8	4	7	3	1	5	1	8									
1528	1602	8	CHARA... spaces f...																	
1536	1610	14	CIYX Sun...	6	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	X
1538	1624	8	A13 CIYX Sun...	-	-	-	-	*	*	*	*									
1558	1632	8	CHARA... spaces f...																	
1566	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 CIYX Keyfob	2	7	5	1													
1622	1696	4	P05H Prop 65 ...																	
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 EPP 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 – EPP TASK Shown:

The screenshot shows the 'GSIP Defect' application window. At the top, it displays 'Machine Code: 25327' and 'Defects: 2'. Below this, 'PVI: 210000006' and 'CSN: <Not Queued>' are shown. A large 'BC' logo is visible. The main area shows 'C2-BC-004R' and 'TASK: 10242'. Below that, 'COOLANT HOSE CONNECTION SCAN RH' and '25-C2-266L' are displayed. At the bottom, a 'Defect Data Viewer' table is shown:

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 the text: '?? < Can't locate via Machine Code >' and 'DD-SS-000A'.





Example – EPP 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 /
Keypress 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 EPP 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	T1M160440R2655620	...

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.



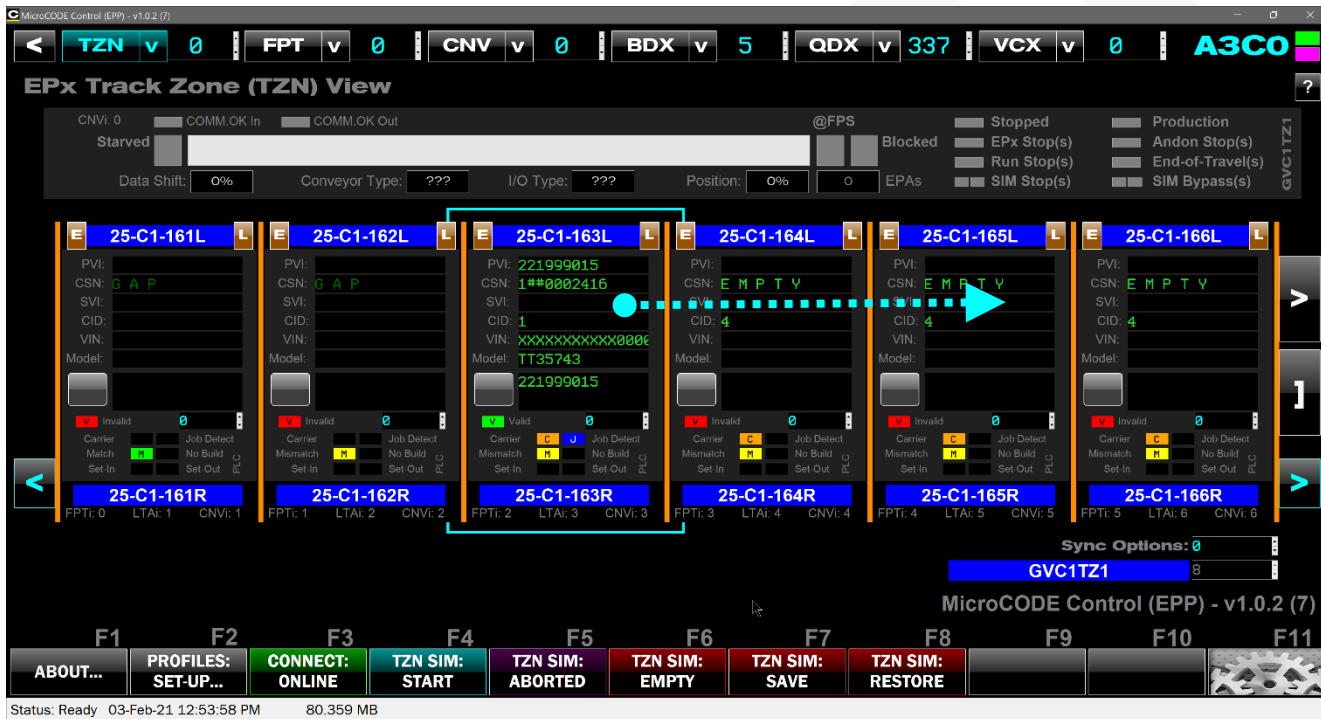


5) The JOB FOCUS now follows the PVI

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.





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

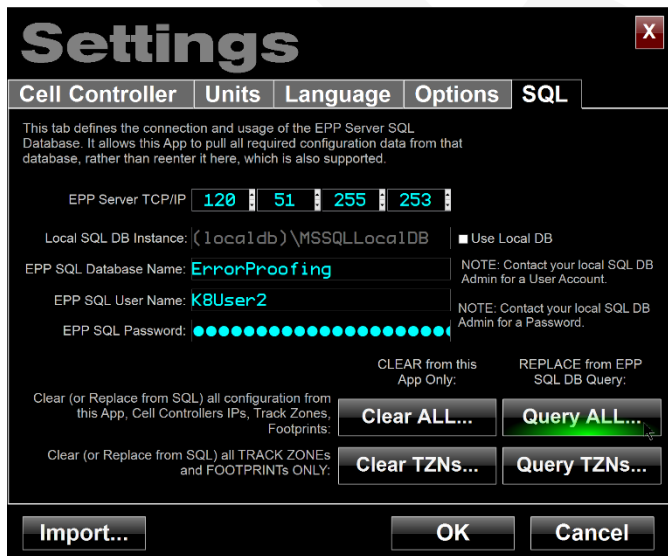
For SEP this includes:

- GEPICS Format for each Cell Controller

Implementation: When the App re-queries the EPP 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.

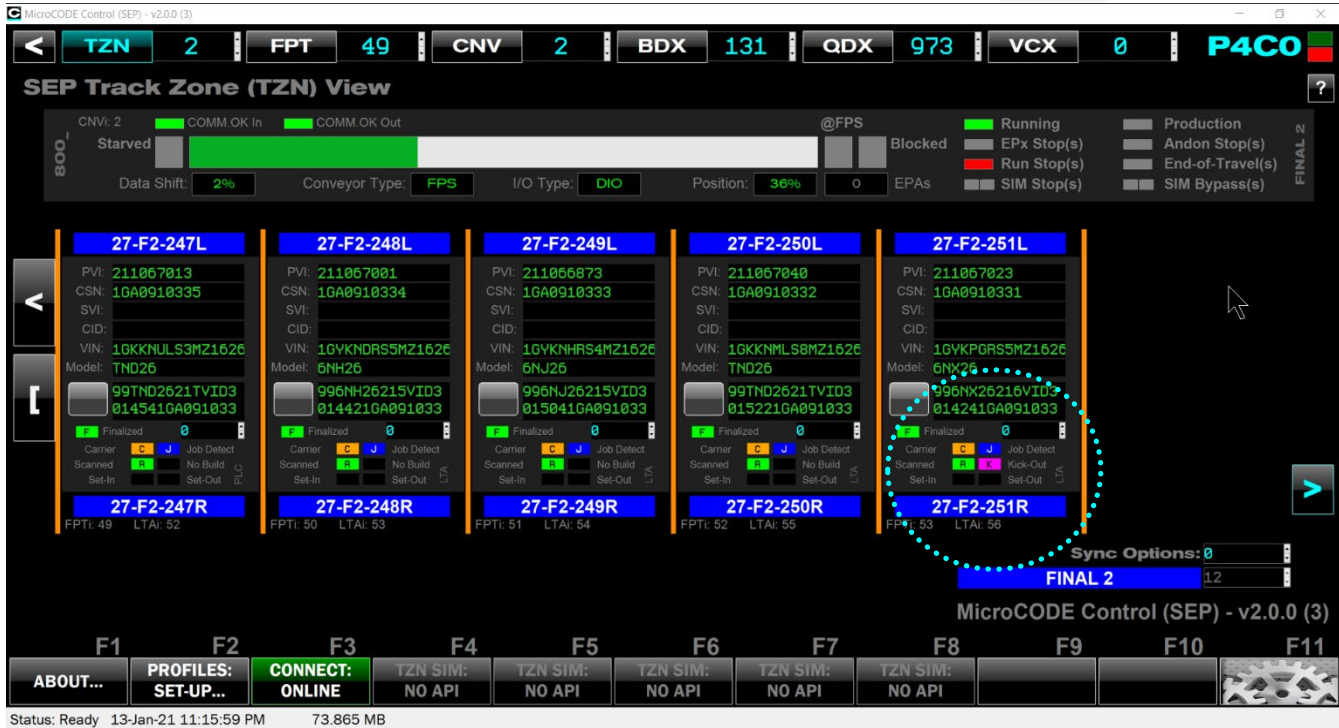




7) Display 'Kick-Out' Jobs

Requirement: SEP has the ability to KICK-OUT a Job.
 This action stops work on the Job but does not open GSP 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 v1.0.2 a (7)

The following defects were fixed in **MicroCODE Control (EPP)** 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.

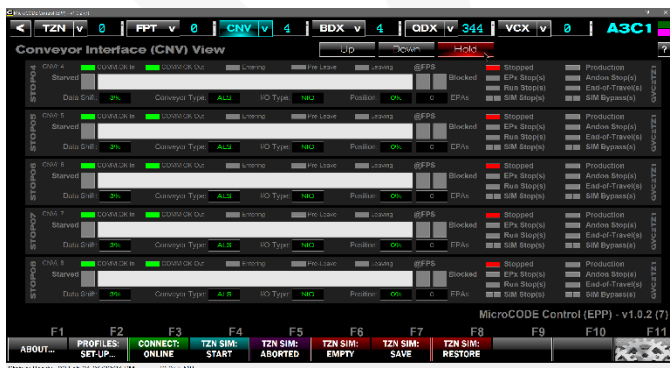
Issue: There were accidentally limitations in the App when dealing with more than (16) Conveyor interfaces.

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 v1.0.2 a (8)

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

1) Display TASK Information on EPP Trace Queue Events

Requirement: The EPP 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.

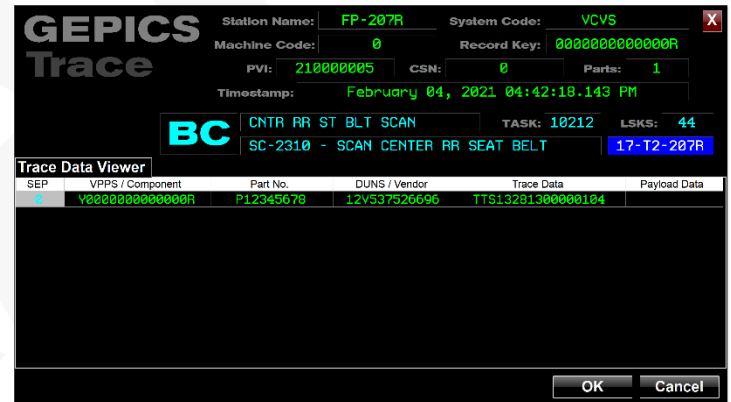
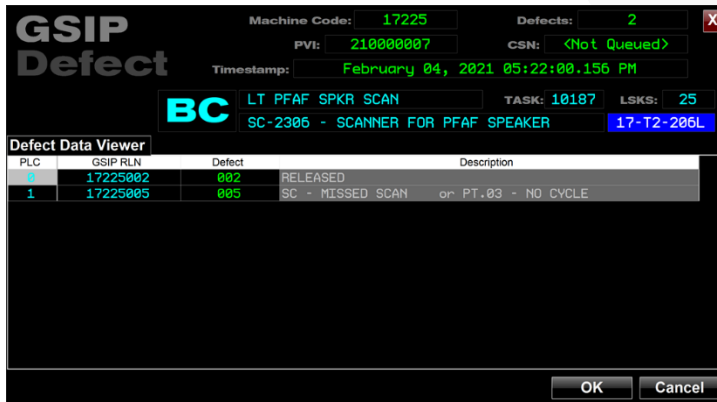
This allows for a backward link to the EPA that generated the TRACE or DEFECT.

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

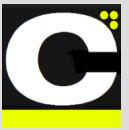
But, in EPP, the Machine Code was eliminated from the TRACE Queue, preventing this backward link, so when the Machine Code is missing the Control App uses the **Station Name** to locate the TASK that mostly likely generated the TRACE Data.

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 EPP TRACE Queue there is nothing available to backward link to a unique TASK.



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



Corrected in Version v1.0.2 a (8)

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

1) The Asynchronous Entering, Leaving, were not working with the new EPP AGV indexed code

Issue: Due to EPP development that occurred after the release of this App the App's interface to the AGV PLC code was not working.

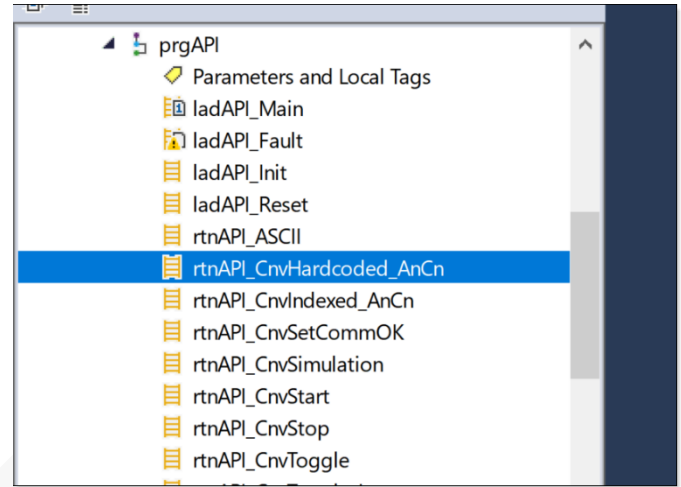
Solution: This has been rectified but also requires the EPP AGV Code be CDE generated after February 5, 2021.

Internal to the App the generation of the ALS and AGV simulated conveyor interlocks have been separated to remain compatible with the two different version of the EPP PLC code.

Also, the App now reads TAKT % Travel directly from the Footprint Programs (FPNNN) and not from the Conveyor interface Tags. EPP—like SEP—generates its own % Travel for Stop Stations and ignores the .POS return by the Conveyor.

NOTE: There is also a difference in the MicroCODE Control App's API code that is inserted into the EPP PLC.

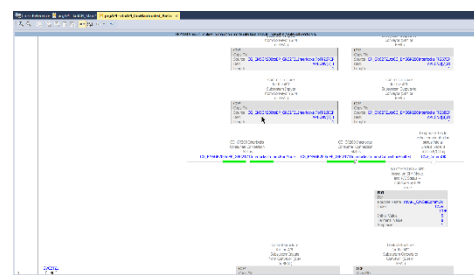
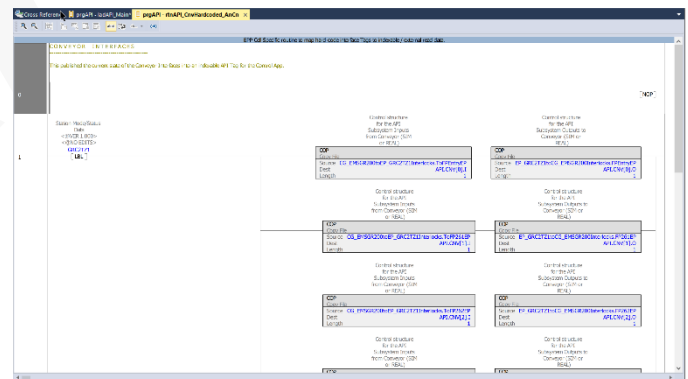
For an **ALS** Conveyor use the 'hard-coded' Footprint name Tag version:



This routine is called with a simple JSR for all Conveyor Interfaces...

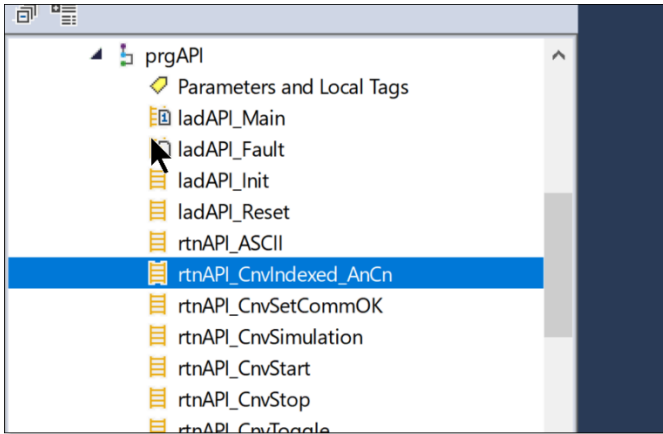


A single rung handles all the named Conveyor Interfaces...

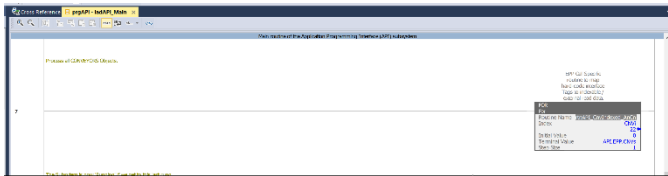




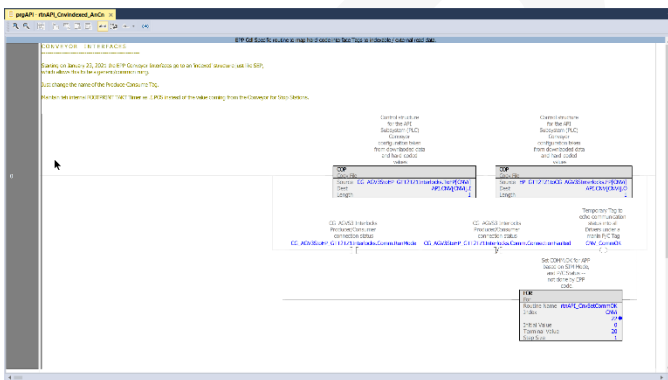
For an **AGV** Conveyor use the 'indexed' Footprint position Tag version:



This routine is called in a FOR-LOOP for all Conveyor Interfaces...



A single rung handles all the indexed Conveyor Interfaces...





New in Version v1.0.3 b (1)

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

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

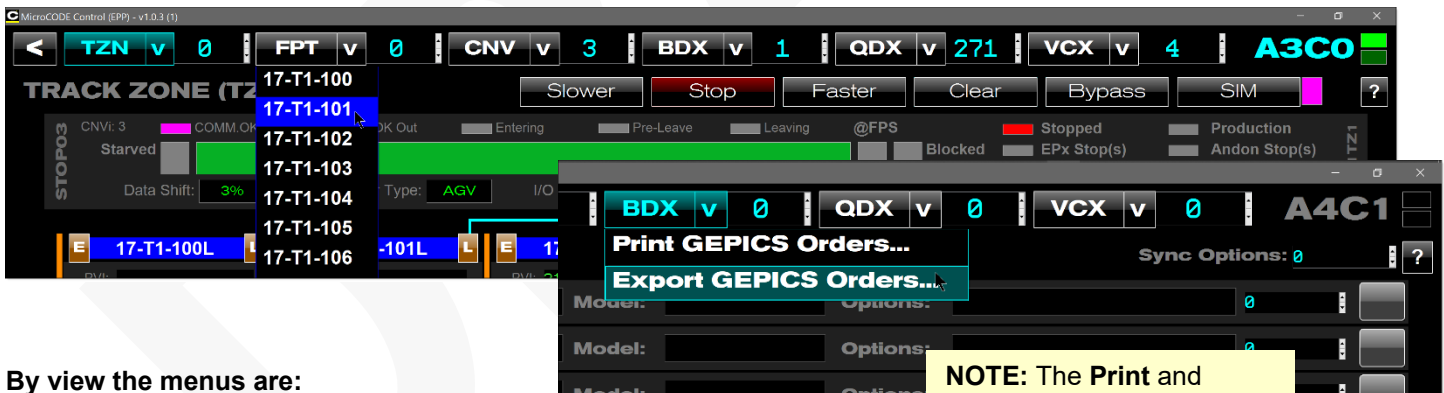
Requirement: User are asking for extensions to the App's functionality, in order to add functionality, the App needs a common method to extend each View.

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.





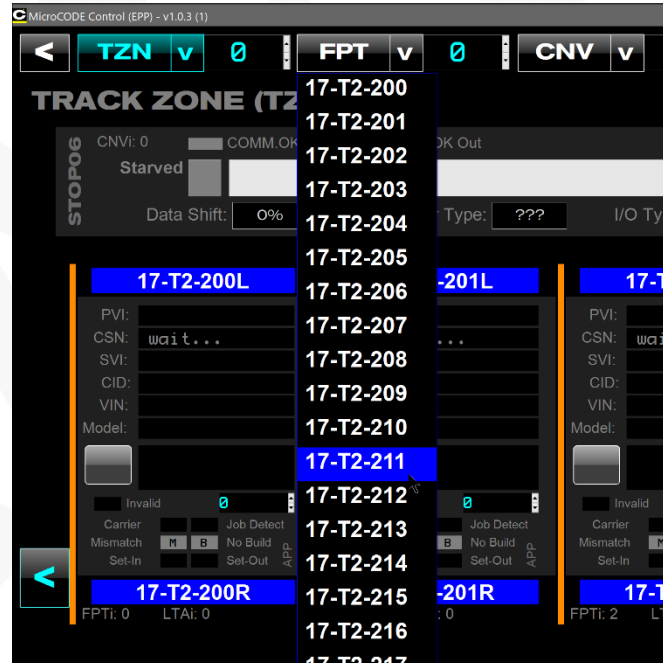
2) The new Pulldown Menus are be used to navigate to any Track Zone and Footprint from the TRACK ZONE View

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.





Corrected in Version v1.0.3 b (1)

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

1) The GEPICS Viewer was not displaying data beyond the Header

Issue: When determining the size of the size of the GEPICS format the v1.0.2 App was not taking the size of the AREA into account. Starting in v1.0.2 the App now reads these Format files to determine that actual size of the data in the PLC, as it variable. But v1.0.2 was not calculating this correcting.

Solution: The App now totals the Area and Zone sizes correctly.

GEPICS Format: **EPP_TRIM, EPP_1022** Items: **323** PVI: **210000019** CSN: **1GA0000019** Source: **PLC**

PLC	GEPICS	Length	Data Item	Description	0	1	2	3	4	5	6	7	8	9
38	38	7	MODEL7		T	T	3	5	7	2	6			
45	45	1	CHARACTER...	Pad Model to length of 8										
46	46	2		Hardcoded CSN Prefix	G	A								
48	48	2	MODEL_YEAR	Last 2 of Model Year	1	9								
50	50	4	CHARACTER...	Resrv for Powertrain Model Cod										
54	54	17	VIN		W	0	V	F	8	6	E	0	X	2
71	71	1	CHARACTER...	Pad VIN to length of 18										
72	72	4		Format Version Number	0	0	0	1						
76	76	4		Earliest Download Supported	0	2	1	2						
80	80	8	Trlr Hitch	EPP_Part Number Trailer Hitch	1	2	3	4	5	6	7	8		
88	88	8	PFAF Spkr	EPP_Part Number PFAF Speaker	8	4	8	1	2	7	1	3		
96	96	8	EBoost	EPP_Part Number E-Boost	8	7	6	5	4	3	2	1		
104	104	8	Temp	EPP_Part Number Temp Number	1	5	0	1	7	7	2	5		
112	112	8	Trace	EPP_Part Number Trace	1	1	5	4	6	3	8	9		
120	120	8	Brk Pd1	EPP_Part Number Brake Pedal	1	3	5	2	5	2	8	6		
128	128	8	Acc Pd1	EPP_Part Number Acceleration P	1	1	5	8	8	7	1	1		

Buttons: Make Seed... Import Seed... OK Cancel





New in Version v1.0.3 b (2)

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

1) Added support for new EPP Conveyor Interface Type: Sub-Assembly Interface (SAI)

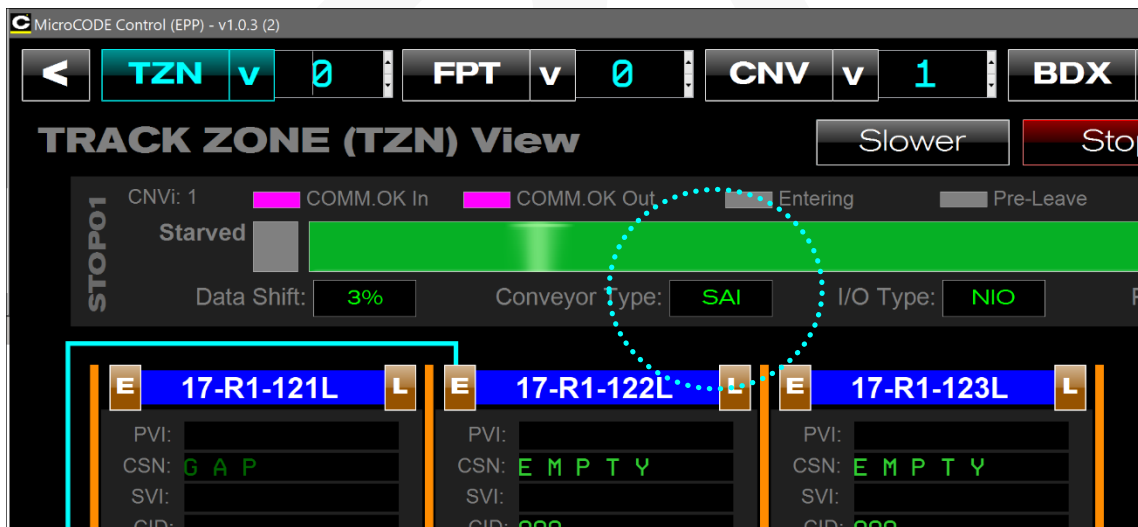
Requirement: In order to handle a Sub-Assembly EPP requires a L85 PLC with a Track Zone. This Track Zone is 'virtual' and does not have a real conveyor associated with it. To handle this EPP has created a new Conveyor type, the "SAI". This type is now recognized by the Control App prior to this release.

Implementation: Two changes were made:

- 1) The App was updated to recognize this new Conveyor Interface type. (SAI).
- 2) The PLC API for the App was updated to recognize this Conveyor Type.

In order to use this new support:

- 1) Update the App.
- 2) Update the API in the affected EPP PLCs.
- 3) In the App, re-query your EPP SQL Database to update the Configuration in the App.





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 'EPP Cell Controller (SITE) View' interface. At the top, there are navigation buttons for TZN, FPT, CNV, BDX, QDX, and VCX, along with a status indicator 'A3C2'. The main area is divided into sections for hardware configuration, including 'ControlLogix Hardware (EMP.CLX)'. A yellow warning banner is displayed at the bottom of the interface, stating: 'Operator: The selected TRACK ZONE is UNAVAILABLE because the selected EPP PLC is OFFLINE.' Below the warning, there are several status buttons: 'PROFILES: SET-UP...', 'CONNECT: ONLINE', 'TZN SIM: RUNNING', 'TZN SIM: ABORT', 'TZN CTL: CLEAR', 'TZN CTL: SAVE', and 'TZN CTL: RESTORE'. A 'CLEAR EVENT' button is also present. The status bar at the very bottom shows '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.

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.





Corrected in Version v1.0.3 b (2)

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

1) The GEPICS Viewer was not displaying all ODD Build Data from the 'Capture'

Issue: When determining the size of the size of the GEPICS format the v1.0.2 App was not taking the size of the AREA into account. Starting in v1.0.2 the App now reads these Format files to determine that actual size of the data in the PLC, as it variable. But v1.0.2 was not calculating this correcting.

Solution: The App now totals the Area and Zone sizes correctly.

GEPICS Build Data Viewer
 Format: EPP_TRIM, EPP_1007 Items: 329
 PVI: CSN: 1GA0823995 Source: PLC

PLC	GEPICS	Length	Data Item	Description	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
684	684	4	OF_AL...	A11 RPOs	V	P	H														
688	688	4	OF_AL...	A11 RPOs	V	R	G														
692	692	4	OF_AL...	A11 RPOs	V	R	H														
696	696	4	OF_AL...	A11 RPOs	V	R	K														
700	700	4	OF_AL...	A11 RPOs	V	R	L														
704	704	4	OF_AL...	A11 RPOs	V	R	M														
708	708	4	OF_AL...	A11 RPOs	V	R	N														
712	712	4	OF_AL...	A11 RPOs	V	R	R														
716	716	4	OF_AL...	A11 RPOs	V	V	4														
720	720	4	OF_AL...	A11 RPOs	W	J	0														
724	724	4	OF_AL...	A11 RPOs	W	M	K														
728	728	4	OF_AL...	A11 RPOs	W	U	5														
732	732	4	OF_AL...	A11 RPOs	X	L	8														
736	736	4	OF_AL...	A11 RPOs	Y	4	K														
740	740	4	OF_AL...	A11 RPOs																	
744	744	4	OF_AL...	A11 RPOs																	
748	748	4	OF_AL...	A11 RPOs																	
752	752	4	OF_AL...	A11 RPOs																	

1288	1288	8	J01A	EPP_Part...	4	2	6	4	5	1	7	7									
1296	1296	8	V04B	EPP_Part...	4	2	6	9	2	1	8	5									
1304	1304	8	M10	EPP_Part...	4	2	7	0	5	4	2	0									
1312	1312	4	M13	EPP_Part...	0	2	2	2													
1316	1316	8	A07F	EPP_Part...	-	-	-	-	*	*	*	*									
1324	1324	8	A07G	EPP_Part...	-	-	-	*	*	*	*	*									
1332	1332	8	A10A	EPP_Part...	4	2	7	3	8	7	4	9									
1340	1340	8	A01Y	EPP_Part...	4	2	5	2	9	6	2	0									
1348	1348	8	X28C	EPP_Part...	4	2	6	4	6	2	4	0									
1356	1356	8	M03N	EPP_Part...																	
1364	1364	8	J01C	EPP_Part...																	
1372	1372	8	M20X	EPP_Part...																	





2) Corrected display of Stop Station & Travel (TAKT Time) for Track Zones after the 1st one in the EPP PLC

Issue: The Percent (%) Travel / TAKT Time for Stop Stations in Track Zones 2 – 9 were not being read from the proper Footprint program.

This was logging errors...

```

++ Message: Read DINT failed - Tag: PROGRAM:FPNam.ConvStatus.PercentTravel, ErrorCode=CIP Error 0x0004: Malformed tag or tag does not exist,
NOTE: Check PLC for missing TAG, missing DHF, or a software Version issue, ControlLogix IP=[120.1.1.22], Slot=[0]
Event: (see 'Message:' above) Time: Wednesday, February 17, 2021 03:11:39.602 PM
Class: MicroCODE.LogixController Type: Logix.MissingTag

++ Message: Read DINT failed - Tag: PROGRAM:FPNam.ConvStatus.PercentTravel, ErrorCode=CIP Error 0x0004: Malformed tag or tag does not exist,
NOTE: Check PLC for missing TAG, missing DHF, or a software Version issue, ControlLogix IP=[120.1.1.22], Slot=[0]
Event: (see 'Message:' above) Time: Wednesday, February 17, 2021 03:11:43.619 PM
Class: MicroCODE.LogixController Type: Logix.MissingTag

++ Message: Read DINT failed - Tag: PROGRAM:FPNam.ConvStatus.PercentTravel, ErrorCode=CIP Error 0x0004: Malformed tag or tag does not exist,
NOTE: Check PLC for missing TAG, missing DHF, or a software Version issue, ControlLogix IP=[120.1.1.22], Slot=[0]
Event: (see 'Message:' above) Time: Wednesday, February 17, 2021 03:11:47.613 PM
Class: MicroCODE.LogixController Type: Logix.MissingTag
  
```

Solution: The API had to be updated to include 'Footprint Index in track Zone' in the program:

ladAPI_Configuration_AnCn

This must be configured in Rung 2 "Footprints" as shown below. These reset to ZERO at the start of each new Track Zone.

The screenshot shows a ladder logic rung with several MOV instructions. Each MOV instruction is associated with a 'Control structure for the API Subsystem (PLC)'. The instructions are:

- MOV Move, Source 8, Dest API.EPP.FPTs
- MOV Move, Source 0, Dest API.EPP.FPTsi
- SUB Subtract, Source A API.EPP.FPTs, Source B 1, Dest API.EPP.FPTsi
- MOV Move, Source 0, Dest API.FPT[0].TZNI
- MOV Move, Source 0, Dest API.FPT[0].FPTi
- MOV Move, Source 1, Dest API.FPT[0].LTAi
- MOV Move, Source 1, Dest API.FPT[0].CNVi
- MOV Move, Source 0, Dest API.FPT[1].TZNI
- MOV Move, Source 1, Dest API.FPT[1].FPTi
- MOV Move, Source 2, Dest API.FPT[1].LTAi
- MOV Move, Source 2, Dest API.FPT[1].CNVi

NOTE: This change was removed... do not implement. See v1.0.3b4



New in Version v1.0.3 b (3)

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

No new features

Corrected in Version v1.0.3 b (3)

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

1) The EPP PLC does not start work for new Job Data presented in the 1st Footprint

Issue: For any Footprint but the first (or only in this case) you can place a simulated Job in the Footprint before and click [E] to generate the ENTERING Interlocks and work starts on the EPP Tasks.

But, for the first Footprint—or only Footprint—you cannot do this, and so you cannot start the EPP Tasks.

Solution: When generating new Jobs for the first (or Only) Footprint in a Track Zone the App now places the new Job in the FPEnter Buffer and issues the ENTERING Interlocks automatically.

1) The EPP PLC does not shift data out of a Footprint in a SAI Single Footprint Track Zone when given the LEAVING interlocks

Issue: When a GAP is present in a Single Footprint Track one for a Sub-Assembly the EPP PLC does not shift that GAP / BLANK Data into the Tracking Buffer. IN a multi-Footprint Track Zone, it does.

Solution: The Control App now forces this to happen thru its own internal command.





New in Version v1.0.3 b (4)

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

No new features

Corrected in Version v1.0.3 b (4)

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

1) Simulation Commands in the first Footprint were not reliable

Issue: Commands to GAP, EMPTY and create JOB in the First Footprint were not always working.

Solution: Correction #2 in **v1.0.3 b (2)** was an error. This needed to be removed and corrected internal to the App.

2) Conveyor names from the SQL are being overwritten by the name of the first STOP

Issue: After going ONLINE with an EPP PLC the name of the Conveyor associated with a Track Zone is being changed to the name of the first STOP from the PLC.

PLC		Track Zones						
TZNi	Name	Footprints	FPTsi	Conveyor	AREA Format	ZONE Format	?	
0	GVI1TZ1	3	0	STOP01	EPP_TRIM.txt	EPP_1007.txt	Y	
1	GVG1TZ1	1	3	STOP01	EPP_TRIM.txt	EPP_1008.txt	Y	
2	GVI1TZ1	4	4	STOP01	EPP_TRIM.txt	EPP_1009.txt	Y	
3		0	8				N	
4		0	0				N	

Solution: This was not causing any issues but should not have been allowed to happen none the less.

PLC		Track Zones						
TZNi	Name	Footprints	FPTsi	Conveyor	AREA Format	ZONE Format	?	
0	GVG1TZ1	3	0	GVG1_ALS	EPP_TRIM.txt	EPP_1007.txt	Y	
1	GVG1TZ1	1	3	GVG1_ALS	EPP_TRIM.txt	EPP_1008.txt	Y	
2	GVI1TZ1	4	4	GVR1_ALS	EPP_TRIM.txt	EPP_1009.txt	Y	
3		0	8				N	
4		0	0				N	

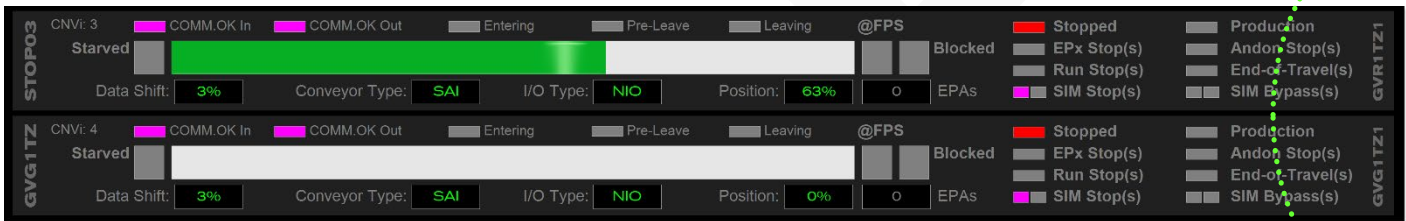


3) Track Zone associated with every Conveyor is not always correct

Issue: Commands to GAP, EMPTY and create JOB in the First Footprint were not always working.



Solution: Correction #2 in this release uncovered this issue in Multi-Track Zone Controllers, along with a few others.

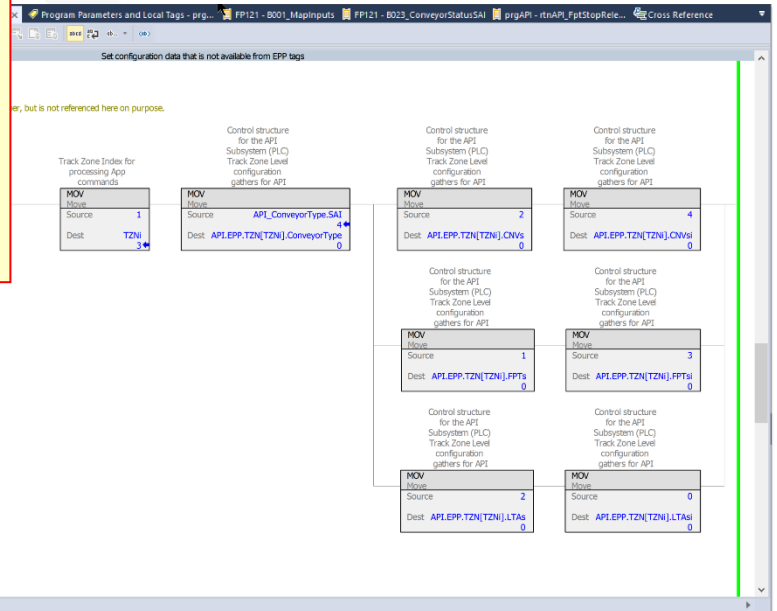


NOTE: Review your TRACK ZONE and FOOTPRINT Configuration rungs in the EPP API very carefully.

CNVsi is accumulative and includes **FPEntry** Conveyor Interface.

FPTsi is accumulative and does not include a space for **FPEntry**.

LTAsi always starts a ZERO and will probably be removed in the future.





New in Version v1.0.3 b (5)

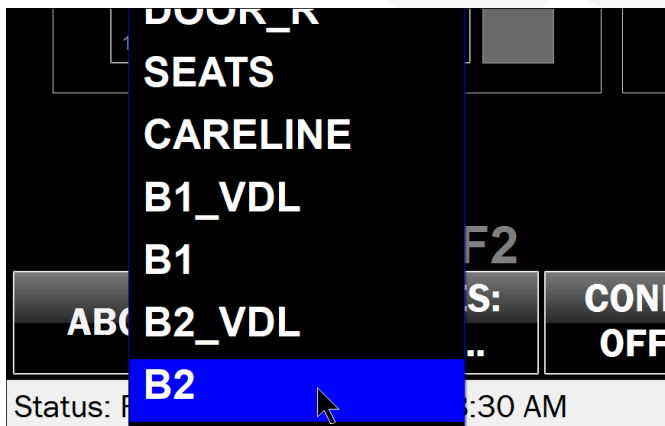
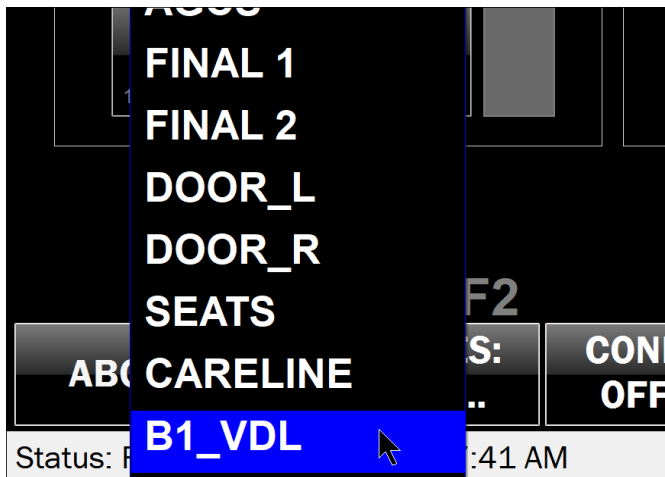
The following features were added to **MicroCODE Control (EPP)** in this Release:

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 outlines in blue, you can use the Mouse Wheel or the Keyboard Arrow Keys to scroll the menu and access all items.





Corrected in Version v1.0.3 b (5)

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 shows 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 display area is titled 'CONVEYOR Interface (CNV) View' and shows a list of conveyor interfaces (CNV) with their status (Starved, Entering, Pre-Leave, Leaving, Blocked, Stopped, Production) and various control buttons (EPx Stop(s), Run Stop(s), SIM Stop(s), Andon Stop(s), End-of-Travel(s), SIM Bypass(s)). The status of the conveyor interfaces is shown as 'Starved'. The bottom status bar shows 'Status: Ready 21-Feb-21 09:30:14 AM 74.236 MB'.





New in Version v1.0.3 b (6)

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

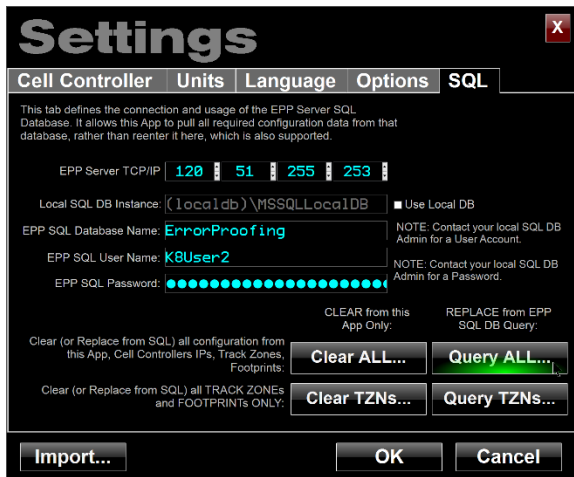
1) SQL Interface has been simplified.

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:

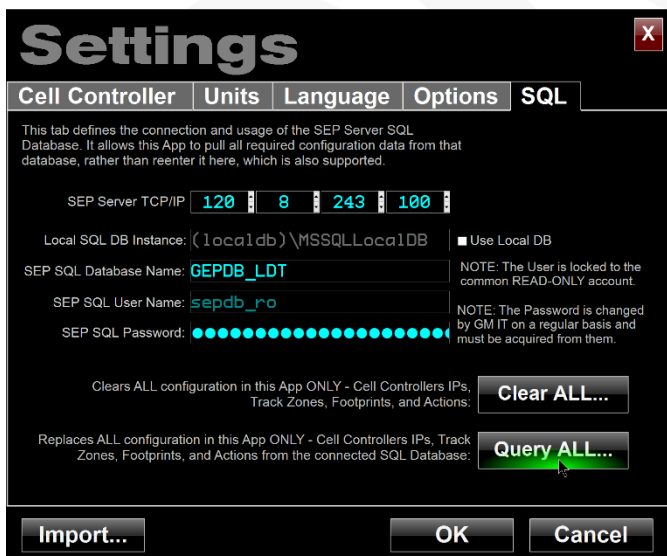


NOTE: The interface to the SQL Database now makes every effort to maintain:

- **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.



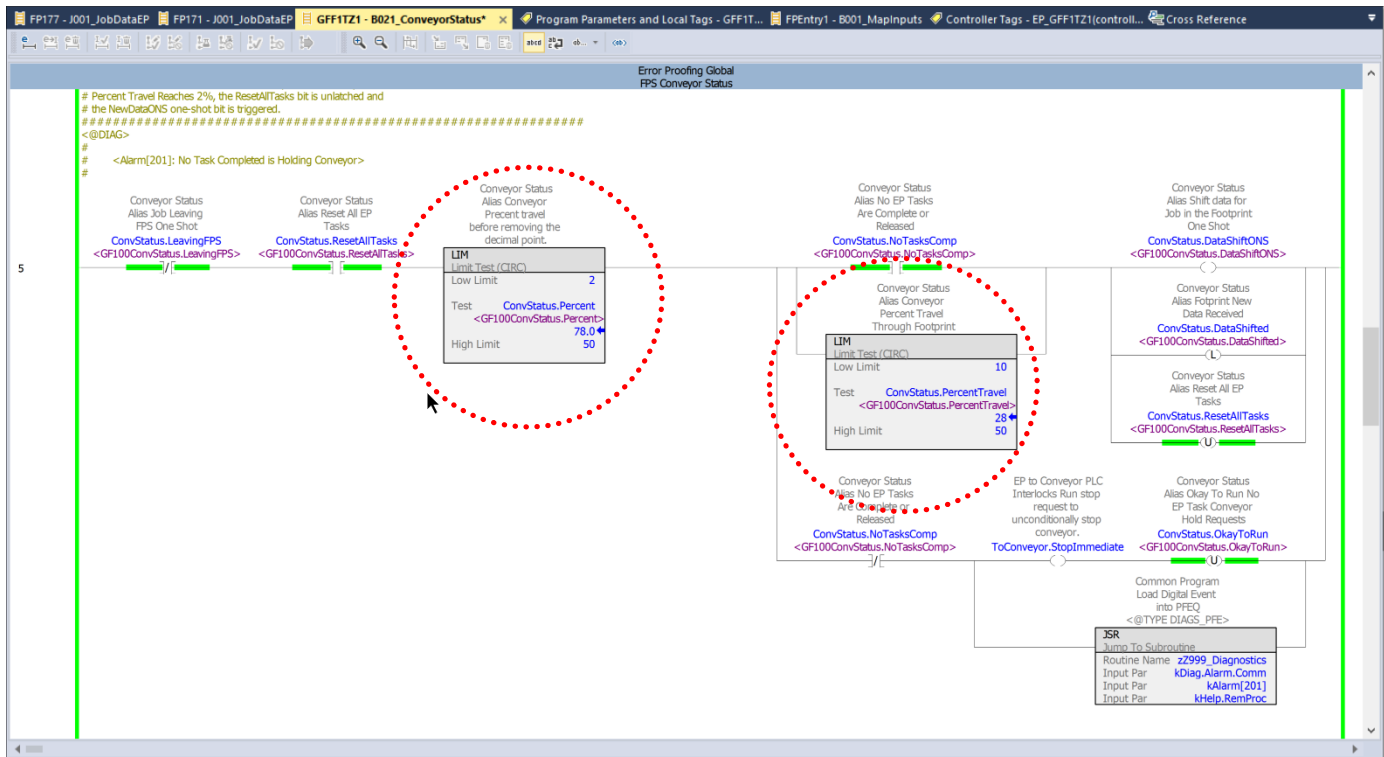


Control.NET™ App for GM EPP

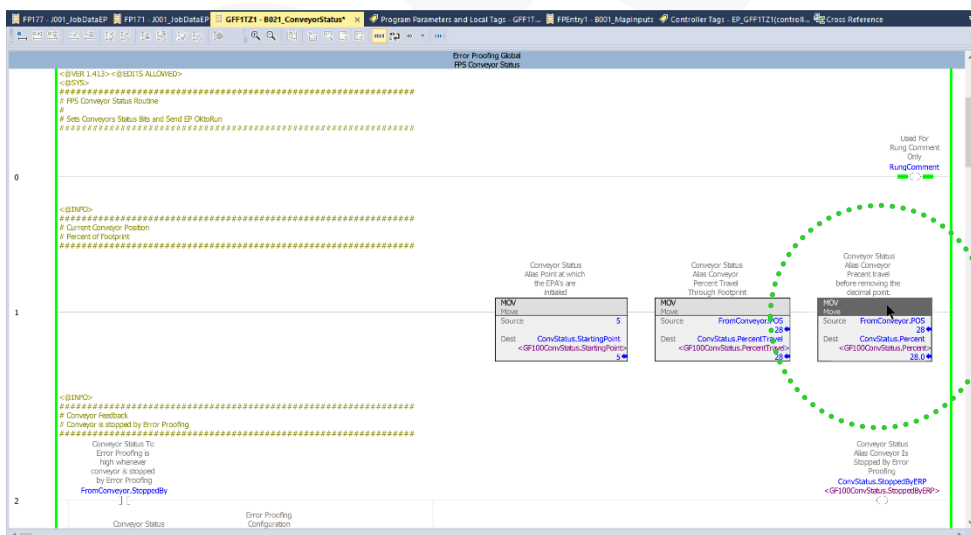
Release Notes Alpha Release: 3.0.0 a (1)



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.



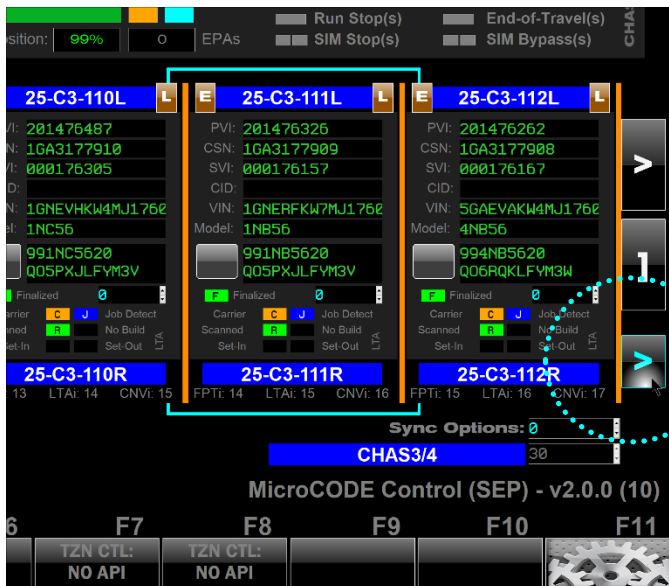
Corrected in Version v1.0.3 b (6)

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 shows '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

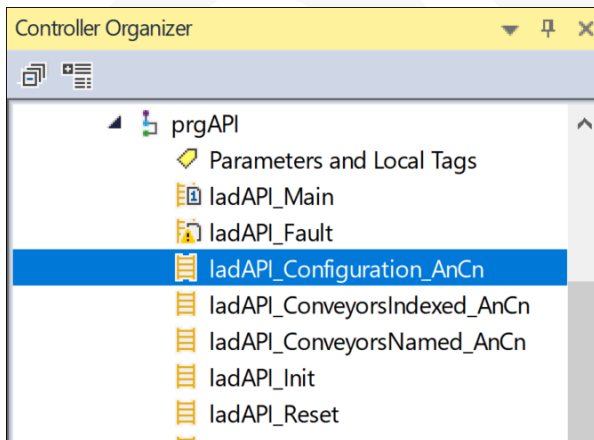
Track Zone #1's Tracking Image is in...
TrackingImage1 *and* **TrackingBuffer1**.

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...

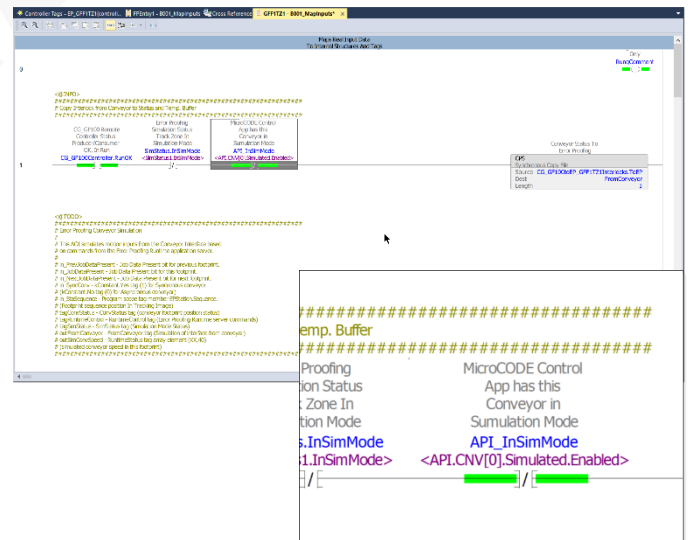
From:

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!" message: "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 parts 25-P1-181R, 25-P1-182R, 25-P1-183R, and 25-P1-184R. Each card shows "Carrier Match" as "Invalid" and "Job Defect" as "No Build". A yellow banner at the bottom of the screen displays a maintenance event: "Maintenance: Communication Error with the AXP-API Interface, ensure MicroCODE 'prgAPI' is installed in Cell Controller's EPP PLC." The banner includes a "CLEAR EVENT" button and a timestamp of [01-Mar-21 02:36:46 PM]. The bottom status bar 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

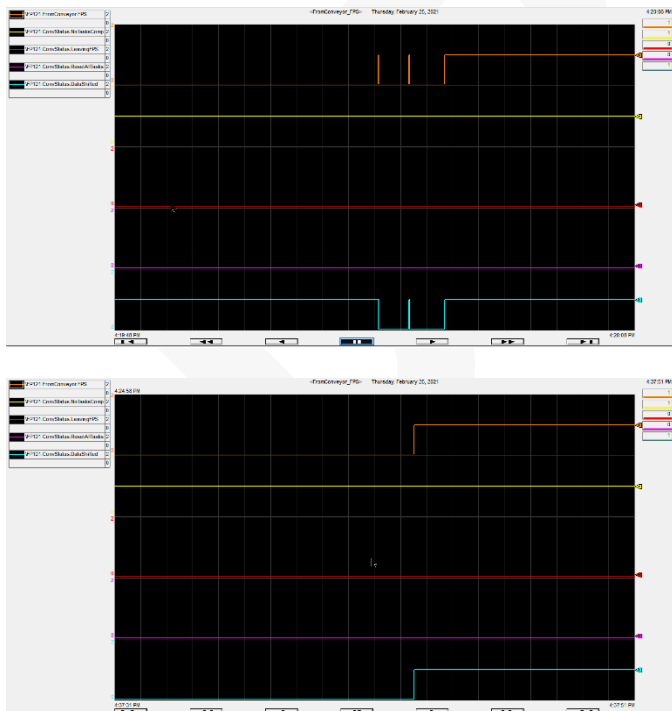
Issue: The App strives to keep your 'Context' consistent from one session to the next, so you can "pick up where you left off". The Conveyor Types were missing from this.

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.

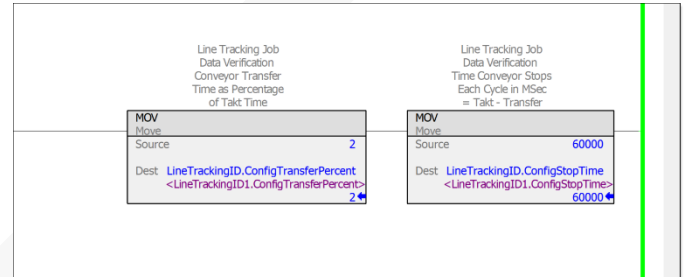
Root Cause: The Jobs were being erased by the EPP PLC code. This was happening in the first Footprint of an AGV/ALS/SAI Track Zone. The App was accidentally given the EPP PLC a 'pulse' in the FPS interlock and this caused two data shifts. Before and After correction are shown below...



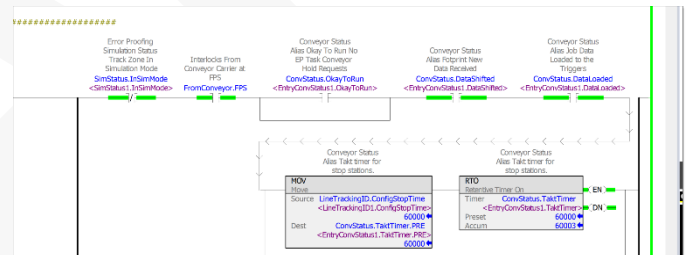
3) TAKT Timer is not running for Footprint

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...

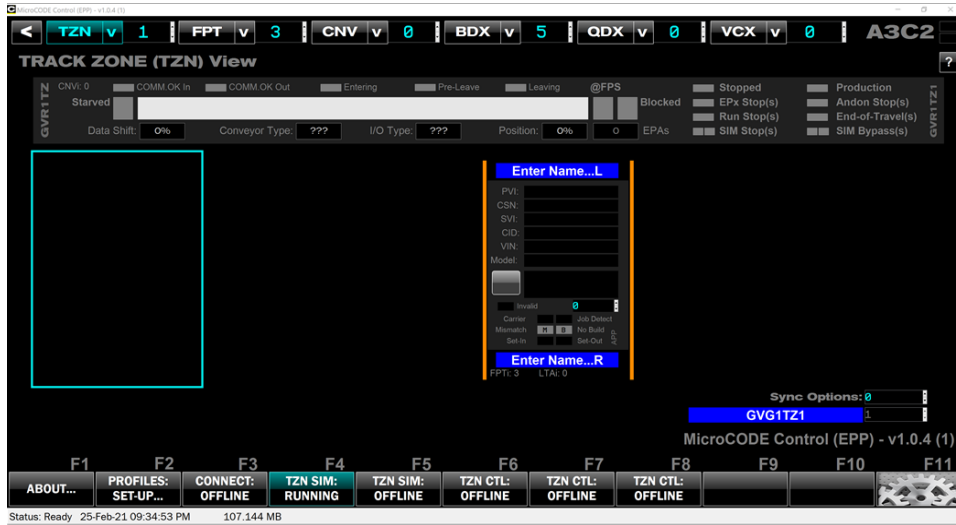




3) User context was being lost when saving or refreshing SQL configuration

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.



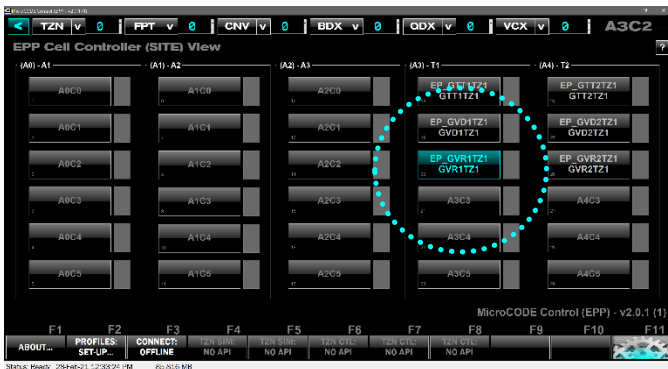


4) Old Controller Reference non-existent after a SQL Import

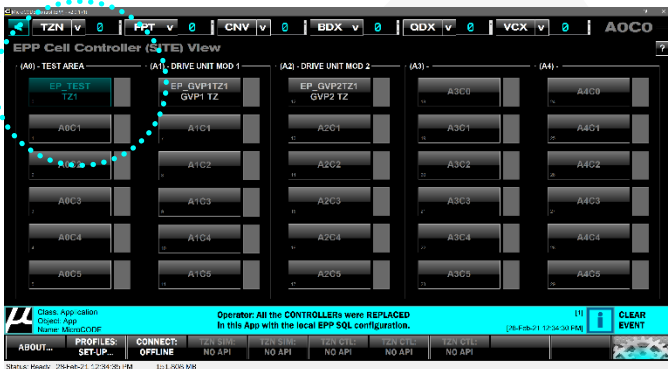
Issue: when switching between EPP SQL DBs the App could be left pointing to a Cell Controller that does not existing in the new SQL DB. This was easily fixed by reselecting an existing Cell but caused confusion.

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 needs to translate EPP Conveyor 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.





Major Test Cases for EPP v2.0.1 b (1) and SEP v2.0.1 b (1)

Fixed Position Stop (FPS)

Task: FPS Conveyor Simulation (SEP)

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

Task: FPS Conveyor Simulation (EPP)

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



Accumulating Lane Stop (ALS)

Task: ALS Conveyor Simulation (SEP)

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

Task: ALS Conveyor Simulation (EPP)

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





Vertically Adjusted Carriers (VAC)

Task: VAC Conveyor Simulation (SEP)

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

Task: VAC Conveyor Simulation (EPP)

- **N/A** – Not supported





Automated Guided Vehicles (AGV)

Task: AGV Conveyor Simulation (SEP)

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

Task: AGV Conveyor Simulation (EPP)

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





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

MicroCODE Control (EPP) - v1.0.4 (1)

TRACK ZONE (TZN) View

Slower Stop Faster Clear Bypass SIM

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

Starved Data Shift: 3% Conveyor Type: SAI I/O Type: SIM Position: 63%

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

Station	PVI	CSN	SVI	CID	VIN	Model	Status
17-11-141L	21000011	1##000011	21000011	11	XXXXXXXXXX21X00	TT35726	Valid
17-11-142L	21000009	1##000009	21000009	9	XXXXXXXXXX21X00	TT35726	Valid
17-11-143L	21000007	1##000007	21000007	7	XXXXXXXXXX21X00	TT35726	Invalid
17-11-144L	21000005	1##000005	21000005	5	XXXXXXXXXX21X00	TT35726	Invalid

Job Restart Empty

MicroCODE Control (EPP) - v1.0.4 (1)

TRACK ZONE (TZN) View

Slower Stop Faster Clear Bypass SIM

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

Starved Data Shift: 3% Conveyor Type: SAI I/O Type: SIM Position: 8%

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

Station	PVI	CSN	SVI	CID	VIN	Model	Status
17-G1-131L	21000002	1##000002	21000002	2	XXXXXXXXXX21X00	TT35726	Valid

Job Restart Empty Gap ODD...

Sync Options: 0

Operator: The Simulated Tracking has created a JOB READY SPACE in the current Track Zone. [25-Feb-21 08:55:11 PM] CLEAR EVENT

Class: Application
Object: App
Name: MicroCODE

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

Status: Ready 25-Feb-21 08:55:17 PM 83.995 MB



New in Version v2.0.1 b (3)

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...

The screenshot shows the 'CONVEYOR Interface (CNV) View' with the following details:

- Navigation:** Buttons for 'Up', 'Down', and 'Hold' are visible at the top right of the interface.
- Conveyor Units:** Five units are listed from CNV: 0 to CNV: 4. Each unit has a 'Starved' indicator and a progress bar.
 - CNV: 0, 1, 2, 3: Progress bars are mostly green, indicating they are running. Position is 24% for units 0-2 and 99% for unit 3.
 - CNV: 4: Progress bar is mostly green with a blue segment at the start. Position is 99%.
- Status Indicators:** Each unit shows 'COMM.OK In' (green), 'OK to RUN' (green), and 'Entering' (green). 'Pre-Leave' and 'Leaving' are greyed out.
- Control Panel:** Includes 'Blocked' (orange), 'EPAs' (grey), and 'SIM Stop(s)' (grey) indicators.
- Bottom Bar:** Shows function keys F1 through F10. F3 is labeled 'CONNECT: ONLINE'. F4-F7 are labeled 'TZN SIM: NO TZN'. The status bar at the very bottom reads: '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').

Machine Code: 17986 **Defects:** 2

PVI: 211067749 **CSN:** <Not Queued>

Timestamp: March 06, 2021 05:31:39.703 PM

110 PLC Queue Position

TT P7C0:TT004 **EPA:** 4

(SSE-3201) PEDAL ASSEMBLY & HARNESS/I SH 17-CP-005L

PLC	GSIP RLN	Defect	Description
0	17986000	000	
1	17986777	777	QUALITY PASS: ALL WORK COMPLETED

Previous Next OK Cancel



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...

The screenshot shows the 'CONVEYOR Interface (CNV) View' in the MicroCODE Control (EPP) v2.0.1 (3) application. The interface is divided into several sections:

- Top Navigation:** Buttons for 'Up', 'Down', and 'Hold'. A status bar at the top shows 'A1CO' and various conveyor status indicators (TZN, FPT, CNV, BDX, QDX, VCX) with values like 0, 0, 0, 0, 36, 0.
- Conveyor Unit Details:** Five conveyor units (CNV: 0 to CNV: 4) are displayed. Each unit has a 'Starved' indicator, 'Data Shift' (0%), 'Conveyor Type' (ALS), 'I/O Type' (NIO), and 'Position' (24% for CNV: 0-2, 99% for CNV: 3-4). Status indicators include 'COMM.OK In', 'OK to RUN', 'Entering', 'Pre-Leave', and 'Leaving'.
- Control Panel:** A panel on the right side of each unit shows various stop and bypass indicators: 'Stopped', 'Production', 'EPx Stop(s)', 'Andon Stop(s)', 'Run Stop(s)', 'End-of-Travel(s)', 'SIM Stop(s)', and 'SIM Bypass(s)'. A 'Blocked' indicator is also present.
- Bottom Control Panel:** A row of function keys (F1-F11) is visible. F1 is 'ABOUT...', F2 is 'PROFILES: SET-UP...', F3 is 'CONNECT: ONLINE', and F4-F7 are 'TZN SIM: NO TZN'. F8-F10 are empty, and F11 is a graphical icon.
- Status Bar:** At the bottom, it shows 'Status: Ready 04-Mar-21 04:09:30 PM 70.889 MB'.



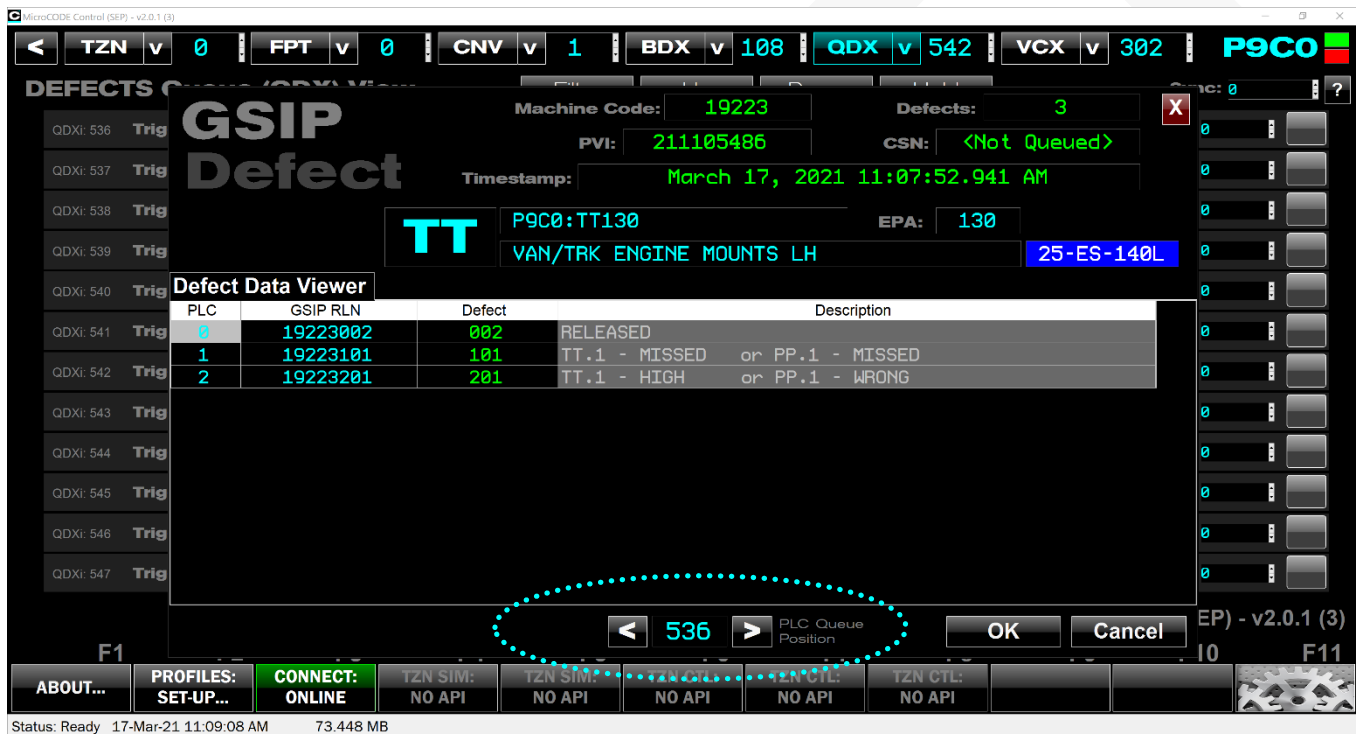


2) You can now move thru the DEFECT, TRACE, and GEPICS Queues from within the Viewers

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).

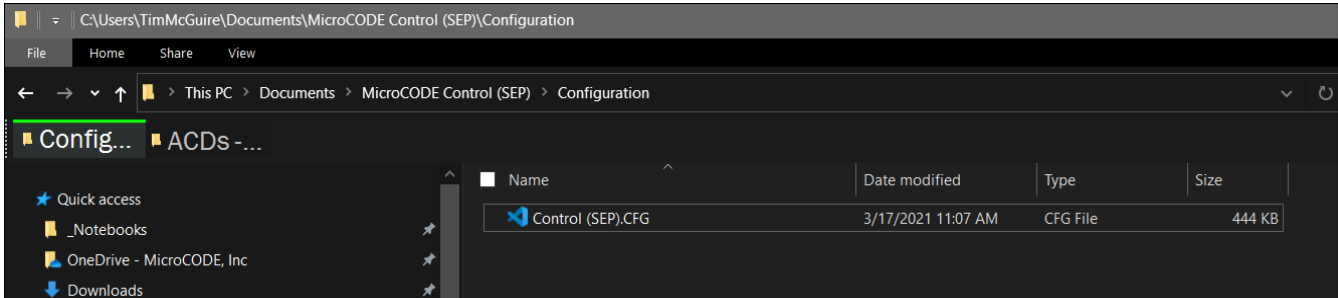
Future versions will allow unlimited walking thru the PLC Queue. Until then close the Viewer and Page UP/DOWN to get the Defects/Traced Components of interest on the Screen first.





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:



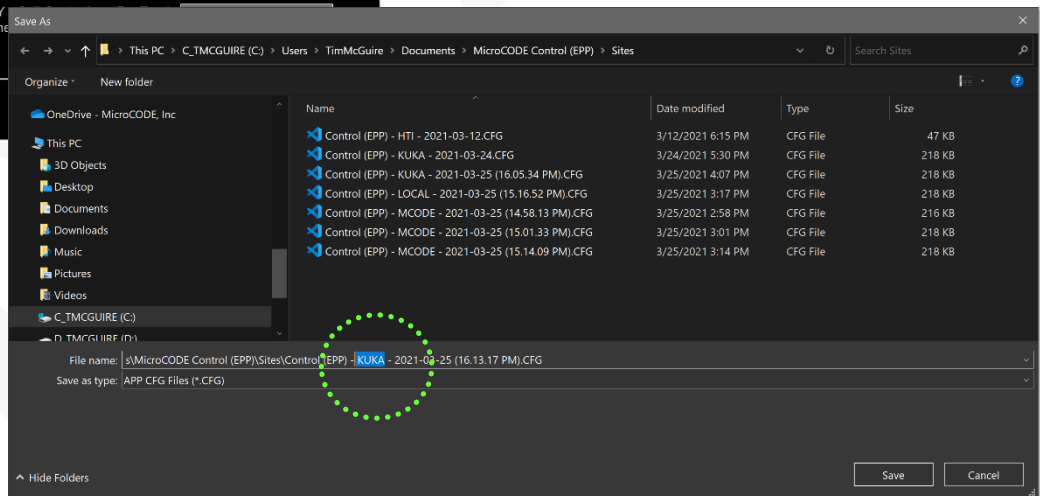
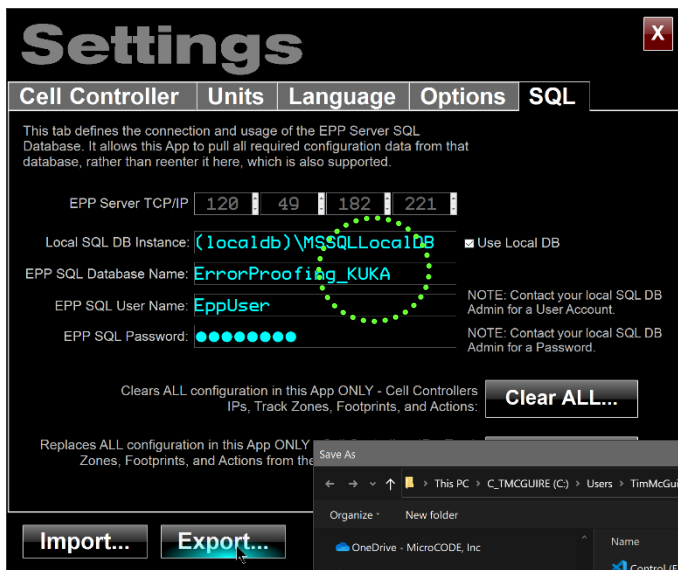
But it was not intuitive. So, we have added an explicit button to do so and that also allows you to give it a specific name in the process for reference:

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).

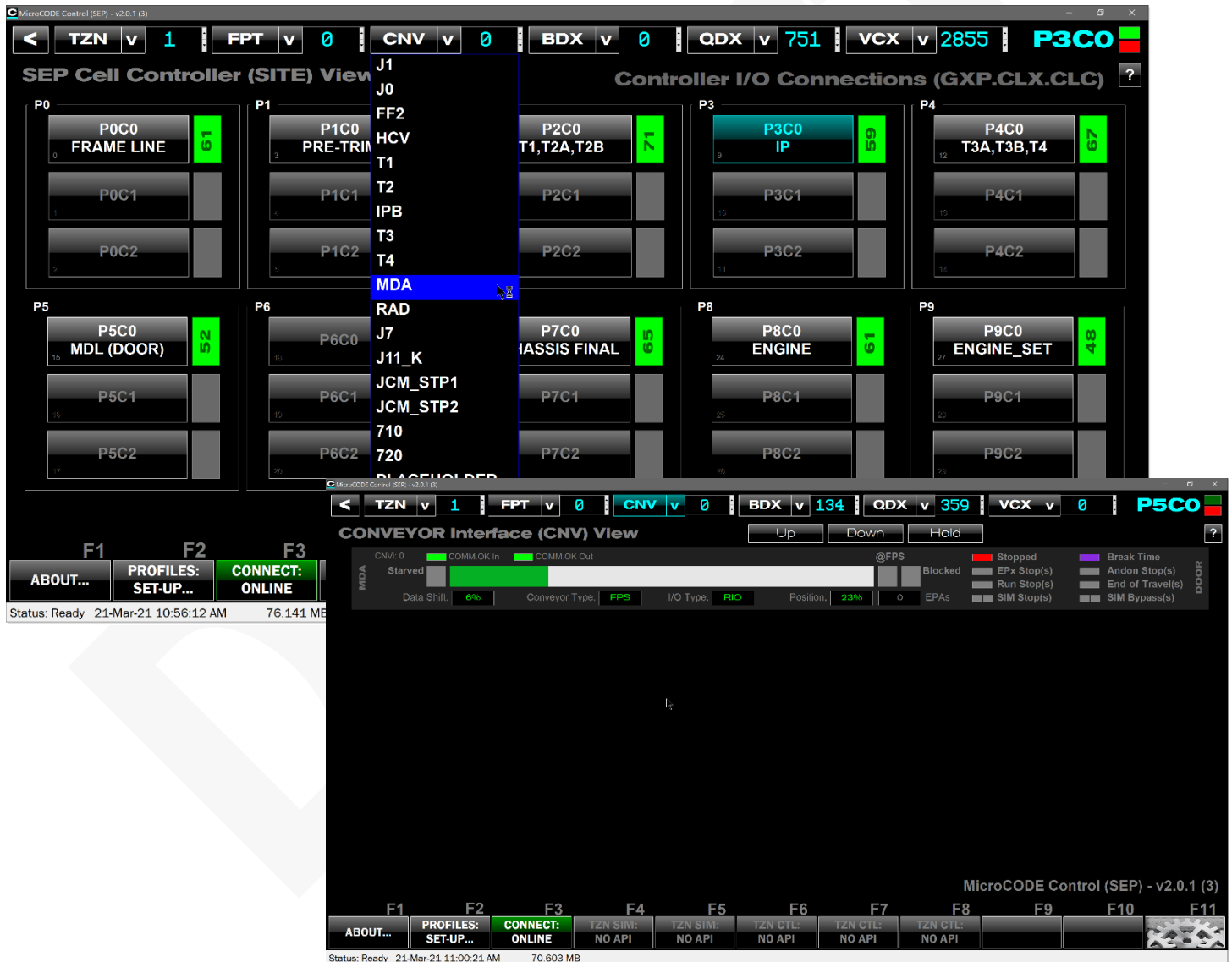
On a Plant radio mechanic people refer to Conveyor names (J0, J1, K11, etc.).

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 two overlapping windows from the MicroCODE Control (SEP) v2.0.1 (3) application.

Top Window: SEP Cell Control
 This window shows a grid of conveyor cells (P0 to P9) and a list of items on the left. The item list includes: 17-T1-001, T1-DC-LH, T1-DC-RH, 17-T2-033, 25-AG-A02, 25-CR-Q, 17-DT-000, 17-DT-000, 25-C1-099, 25-C2-123 (highlighted), 25-C4-173, 25-C5-198, 25-SU-001, 25-TU-001, 25-EL_S, 25-EL-001, 25-C3-150, 27-F1-219, 27-F2-240, 11-ZA-030, 11-ZA-210, 17-T3-053, 17-T4-078, and 17-CP-001. The status at the bottom is "Status: Ready 25-Mar-21 05:09:16".

Bottom Window: TRACK ZONE (TZN) View
 This window shows a detailed view of a specific footprint group, 25-C2-123L. It displays a grid of 12 footprint items (25-C2-123L to 25-C2-128R) with their respective PVT, CSN, SVI, CID, VIN, and Model fields. Below the grid are status indicators for "Incomplete", "Carrier", "Job Detect", "No Read", "Set-In", and "Set-Out". The status at the bottom is "Status: Ready 25-Mar-21 05:11:09 PM 97.169 MB".





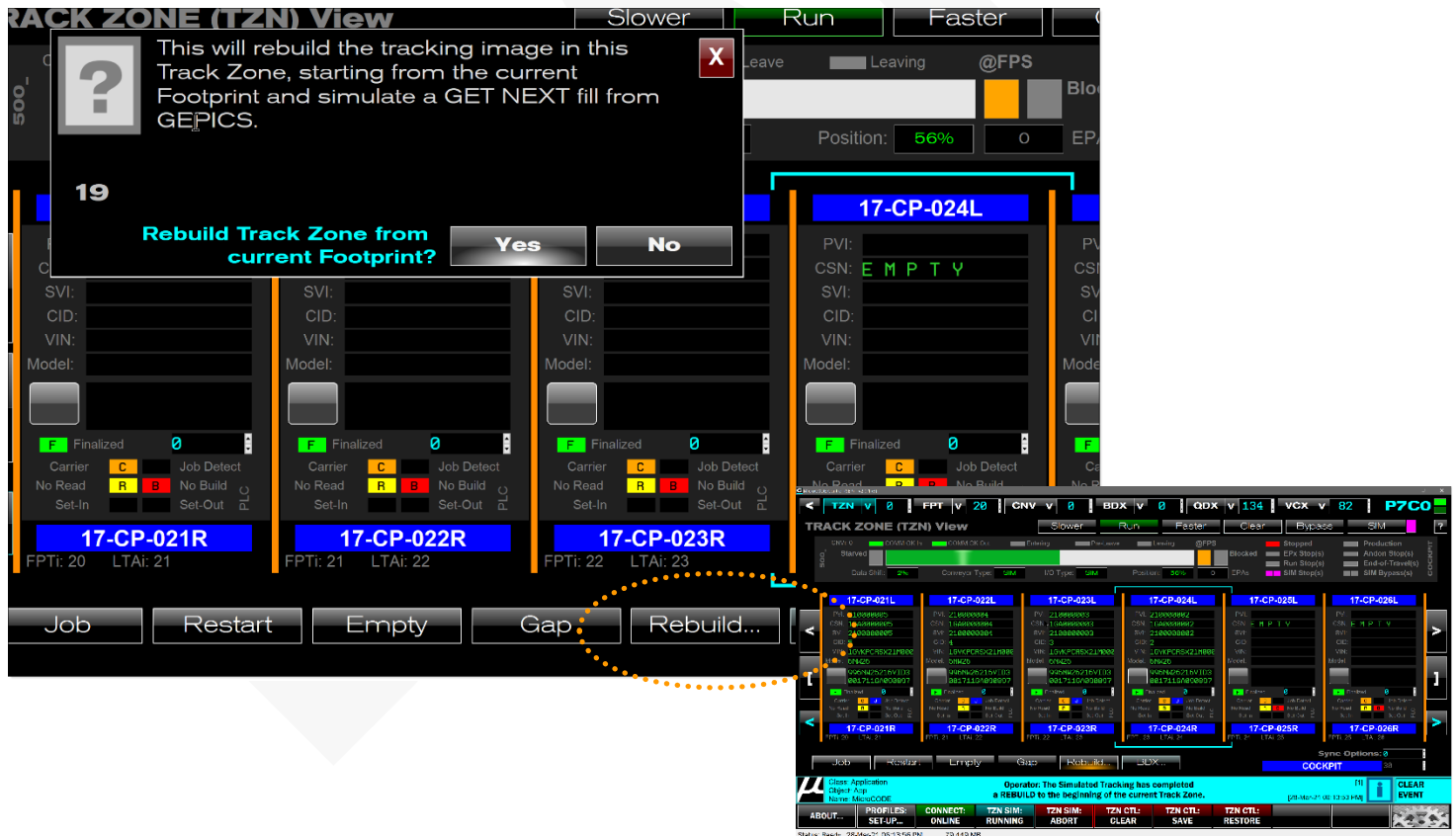
6) ESCAPE KEY is now supported as a BACK button or Screen Closure Command

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.

NOTE: This feature was not completed in time for the v2.01b4 Release but will follow soon afterward in a subsequent Build.





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.

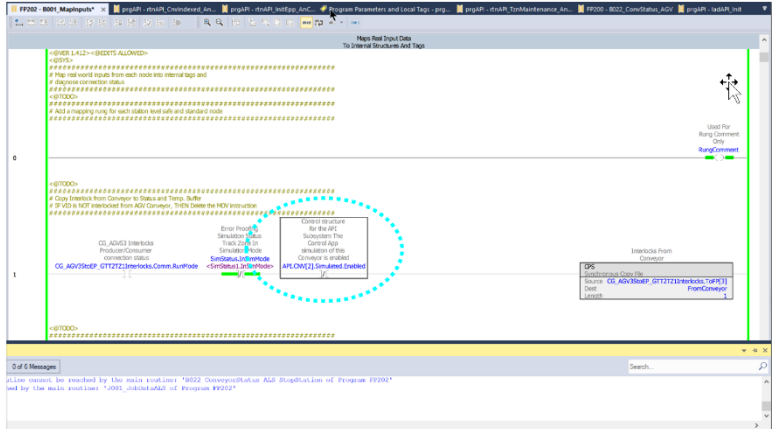


MicroCODE Control – for the GM Scalable Error Proofing (EPP) System

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 (5-6)

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.

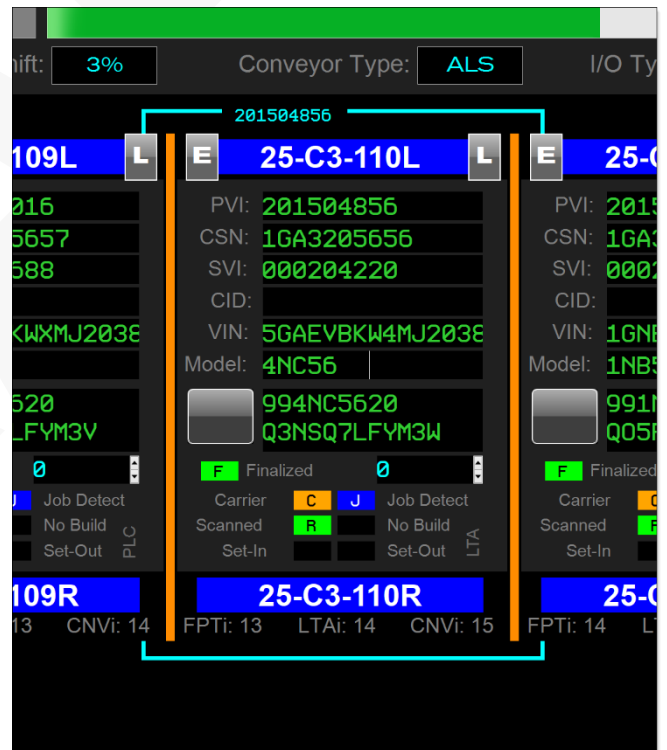
Requirement: make it clear to the User when he has a Job in 'focus' and when he does not.

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:



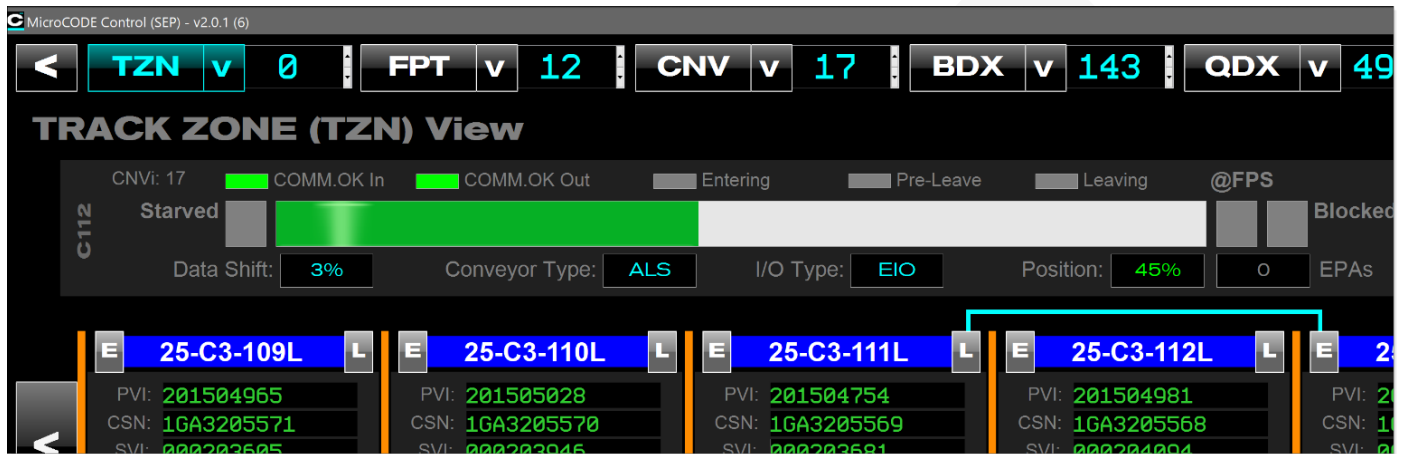


2) Clarify Conveyor Data

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

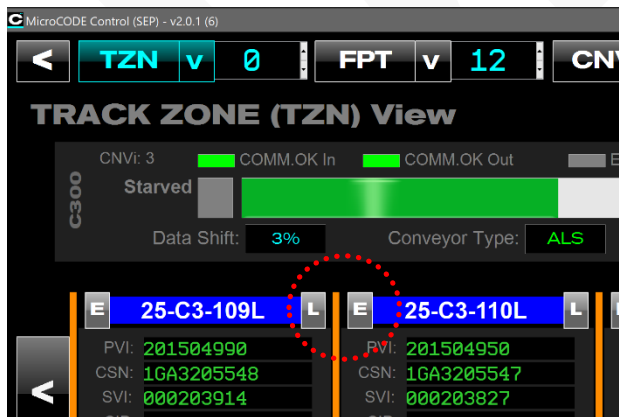


3) Clarify when Entering/Leaving Commands can be used

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 (5-6)

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

1) Issues with simulated FPS Conveyors

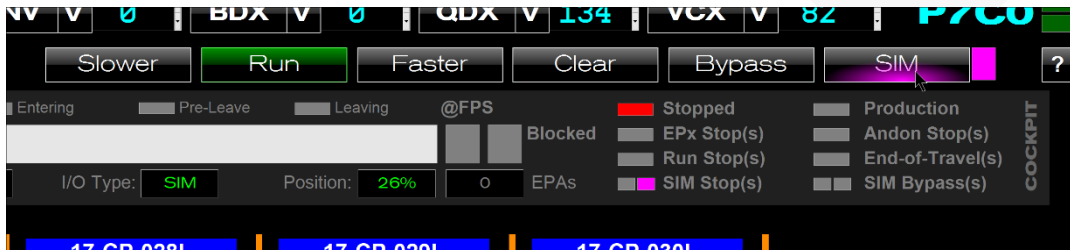
The FPS Conveyor controls were not enabled when in Simulation mode, even though they appeared normal.

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

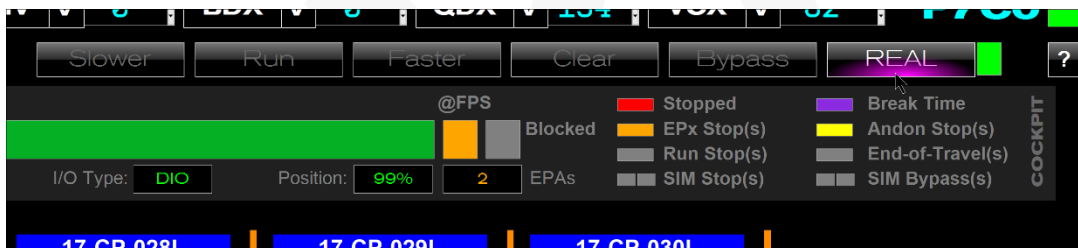
But they were disabled all the time. Whoops. Rushed that Build Release.

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 Stopped Break Time
Blocked EPx Stop(s) Andon Stop(s)
Run Stop(s) End-of-Travel(s)
SIM Stop(s) SIM Bypass(s)

Data Shift: 2% Conveyor Type: FPS I/O Type: DIO Position: 99% 0 EPAs

17-CP-001L PVI: 210000015 CSN: 1GA0000015 SVI: 210000015 CID: 15 VIN: KPCRSX21M0015 Model: 6N425 996N426216VID3 001711GA090897	17-CP-002L PVI: 210000014 CSN: 1GA0000014 SVI: 210000014 CID: 14 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-003L PVI: 210000013 CSN: 1GA0000013 SVI: 210000013 CID: 13 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-004L PVI: 210000012 CSN: 1GA0000012 SVI: 210000012 CID: 12 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-005L PVI: 210000011 CSN: 1GA0000011 SVI: 210000011 CID: 11 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-006L PVI: 210000010 CSN: 1GA0000010 SVI: 210000010 CID: 10 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897
---	---	---	---	---	---

Finalized Job Detect
Carrier No Read No Build
Set-In Set-Out

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

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

Class: Application
Object: App
Name: MicroCODE

Operator: The Simulated Conveyor Simulation has been TOGGLED for the current Track Zone. [25-Mar-21 09:15:56 PM] CLEAR EVENT

TRACK ZONE (TZN) View

Slower Stop Faster Clear Bypass SIM ?

COMM OK In COMM OK Out Entering Pre-Leave Leaving @FPS Running Production
Blocked EPx Stop(s) Andon Stop(s)
Run Stop(s) End-of-Travel(s)
SIM Stop(s) SIM Bypass(s)

Data Shift: 2% Conveyor Type: SIM I/O Type: SIM Position: 90% 0 EPAs

17-CP-005L PVI: 210000009 CSN: 1GA0000009 SVI: 210000009 CID: 9 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-006L PVI: 210000008 CSN: 1GA0000008 SVI: 210000008 CID: 8 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-007L PVI: 210000007 CSN: 1GA0000007 SVI: 210000007 CID: 7 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-008L PVI: 210000006 CSN: 1GA0000006 SVI: 210000006 CID: 6 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-009L PVI: 210000005 CSN: 1GA0000005 SVI: 210000005 CID: 5 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897	17-CP-010L PVI: 210000004 CSN: 1GA0000004 SVI: 210000004 CID: 4 VIN: 1GVKPCRSX21M000 Model: 6N425 996N426216VID3 001711GA090897
--	--	--	--	--	--

Finalized Job Detect
Carrier No Read No Build
Set-In Set-Out

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

Job Restart Empty Gap BDX...

Sync Options: 8

COCKPIT

F1 ABOUT... F2 PROFILES: SET-UP... F3 CONNECT: ONLINE F4 TZN SIM: RUNNING F5 TZN SIM: ABORT F6 TZN CTL: CLEAR F7 TZN CTL: SAVE F8 TZN CTL: RESTORE

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

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



Task: FPS Conveyor Simulation (EPP)

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





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]

DRAFT



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

MicroCODE Control (EPP) - v1.0.4 (1)

← **TZN** v 2 | **FPT** v 4 | **CNV** v 7 | **BDX** v 1 | **QDX** v 381 | **VCX** v 0 | **A3C2**

TRACK ZONE (TZN) View [Slower] [Stop] [Faster] [Clear] [Bypass] [SIM] [?]

CNV: 7 COMM.OK In COMM.OK Out Entering Pre-Leave Leaving @FPS Stopped EPx Stop(s) Production
Starved Blocked Run Stop(s) Andon Stop(s)
Data Shift: 3% Conveyor Type: SAI I/O Type: SIM Position: 63% 0 EPAs SIM Stop(s) SIM Bypass(s) GVR1TZ1

Station	PVI	CSN	SVI	CID	VIN	Model	Valid	Carrier	Job Detect	No Build	Set In	Set Out
17-11-141L	21000011	1##000011	21000011	11	XXXXXXXXXX21X00	TT35726	Valid	C	J	No Build	Set In	Set Out
17-11-142L	21000009	1##000009	21000009	9	XXXXXXXXXX21X00	TT35726	Valid	C	J	No Build	Set In	Set Out
17-11-143L	21000007	1##000007	21000007	7	XXXXXXXXXX21X00	TT35726	Invalid	C	J	No Build	Set In	Set Out
17-11-144L	21000005	1##000005	21000005	5	XXXXXXXXXX21X00	TT35726	Invalid	C	J	No Build	Set In	Set Out

Job Restart Empty

MicroCODE Control (EPP) - v1.0.4 (1)

← **TZN** v 1 | **FPT** v 3 | **CNV** v 5 | **BDX** v 1 | **QDX** v 412 | **VCX** v 0 | **A3C2**

TRACK ZONE (TZN) View [Slower] [Stop] [Faster] [Clear] [Bypass] [SIM] [?]

CNV: 5 COMM.OK In COMM.OK Out Entering Pre-Leave Leaving @FPS Stopped EPx Stop(s) Production
Starved Blocked Run Stop(s) Andon Stop(s)
Data Shift: 3% Conveyor Type: SAI I/O Type: SIM Position: 8% 0 EPAs SIM Stop(s) SIM Bypass(s) GVR1TZ1

Station	PVI	CSN	SVI	CID	VIN	Model	Valid	Carrier	Job Detect	No Build	Set In	Set Out
17-G1-131L	21000002	1##000002	21000002	2	XXXXXXXXXX21X00	TT35726	Valid	C	J	No Build	Set In	Set Out

Job Restart Empty Gap ODD...

Sync Options: 0

GVR1TZ1

Class: Application Operator: The Simulated Tracking has created a JOB READY SPACE in the current Track Zone. [25-Feb-21 08:55:11 PM] CLEAR EVENT

Object: App
Name: MicroCODE

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

Status: Ready 25-Feb-21 08:55:17 PM 83.995 MB





New in Version v2.0.1 b (7)

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

1) Active EPP AREAs are not 'compressed' into AREAs 0-4 in the Control App

In previous versions the EPP AREAs were represented in the exact same positions in the Control App as they were configured in the EPP SQL Database. The EPP SQL Database has no limits (and no definitions) as to what these AREAs represent. The assumed definition was GM Assembly Plant DEPARTMENTS.

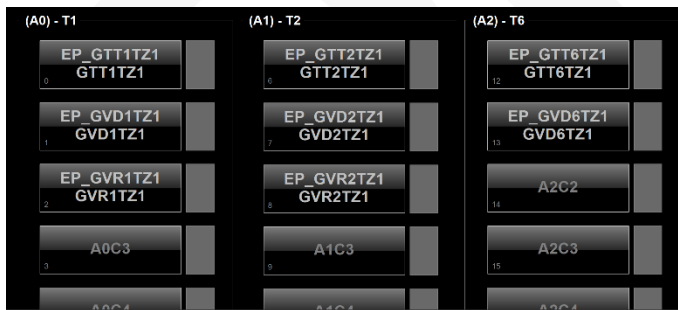
- 11 = Body Shop
- 15 = Paint
- 17 = Hard Trim 1
- 19 = Soft Trim 2
- 25 = Chassis 1
- 26 = Chassis 2 / Engine
- 27 = Final

But, in practice at the GM Strategic Suppliers (GSS) they have been used for:

- Department
- Module
- Section
- Etc.

This has caused the GSS to have more AREAs than the Control App was designed to support.

Example of new behavior: ESYS the AREAs is being used as TRIM Sections, now 'compressed' into AREAs 0 – 2 in the Control App.



Requirement: Support up to five (5) 'AREAs' in Control App with the existing App design.

Implementation: All 'Active' AREAs in the EPP SQL Database are not 'compressed' into AREAs 0-4 in the Control App representation. This has no effect on the interaction of the App with the EPP PLCs.

For reference here is how all the S.S. have EPP AREAs/MODs configured:

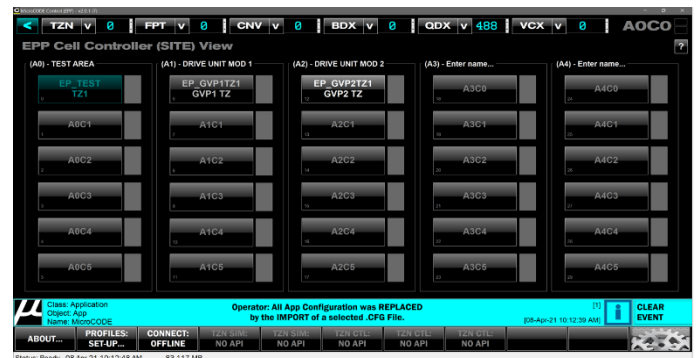
KUKA – AREAs are used as **DEPARTMENTS**.



ESYS – AREAs are used as **SECTIONS**. Note: MOD 6 is actually in AREA 5 in the EPP SQL (off screen)



HTI – AREAs are used as **MODULES**.





New in Version v2.0.1 b (8)

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

1) Support for Single-Sided Track Zones

In order to support Footprints with more than 32 Actions (Tasks) or Single-Sided Track Zones (like DOOR Lines) EPP Users are configuring Footprints with “L” or “R” in the OPERATION Number in the EPP U.I. The Operation Number should be a 3-Digit Number.

Requirement: Recognize and support the use of “L” or “R” in the Operation Number as an indication of LEFT-SIDE ONLY, or RIGHT-SIDE ONLY. Prior to this Release the App showed LL LR and RL RR for Footprint Operation Suffixes.



Implementation: The EPP U.I. and Database should support more than 32 Tasks in a Single Footprint without the Users having to resort to creating fake Footprint names and therefore having to modify the standard PLC code to match.

The EPP U.I. and Database should also support “Single-Sided” Track Zones (Left Only and Right-Only) to properly support Sub-Assembly Build Stations and DOOR Lines.

To remove some confusion the Control App will no longer append the “L” or “R” if the EPP User has already added these suffixes to the Operation number.

Additionally, the App now hides the opposite LH or RH Placard if the User has appended the “L” or “R” to the Operation number.

LH Only Track Zone:



RH Only Track Zone:



Normal Track Zone:





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.

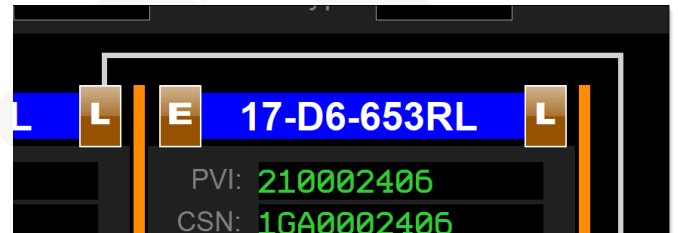
Solution: Immediately after import a new Seed Job it is display in the GEPICS Build Data Viewer.

PLC	GEPICS	Length	Data Item	Description	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	6		PVI		2	1	0	0	0	2	4	0	5								
9	1		CHARA...	Pad PVI ...																	
10	6		BSEQNUM	Last 6 o...																	
16	10		CHARA...	Pad SVI ...	2	1	0	0	0	2	4	0	5								
26	11		CSN		1	G	A	0	0	0	2	4	0	5							
37	1		CHARA...	Pad CSN ...																	
38	7		MODEL7		1	G	1	F	V	6	S										
45	1		CHARA...	Pad Mode...																	
46	2			Hardcode...																	
48	2		MODEL...	Last 2 o...																	
50	4		CHARA...	Resrv fo...																	
54	17		VIN																		
71	1		CHARA...	Pad VIN ...																	
72	4			Format V...																	
76	4			Earliest...																	
80	8		M13	EPP_Pant...																	
88	8		M14	EPP_Pant...																	

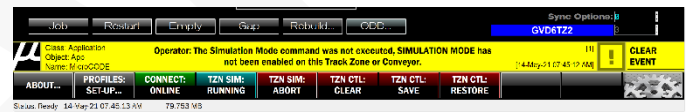
2) Unable to use SIM Commands on AGV

After upgrading to v2.0.1b7 some Stop Stations Simulation Commands were not working.

Issue: With a simulation running on an AGV Track Zone the user was unable to issues Entering/Leaving Commands. The mode and commands were all disabled, even though the Entering/Leaving Commands appeared enabled.



Issues an Entering or Leaving command showed:



The problem was unique Footprint Program names that included the “L” or “R” to support “Single-Sided” Track Zones. This is now supported by the App.

Solution: The App is now aware of the use of “L” and “R” appended to the Footprint names to indicate ‘Left-Side Only’ or ‘Right-Side Only’ and expects the PLC programs names to follow suit.

3) Could not place Stop Stations in SIM

Stop Stations (ALS, AGV, etc.) beyond Track Zone 0 could not be placed in SIM Mode.

Issue: The App was not initializing the Conveyor Ending Index (CNVei) for Track Zones 1,-8 properly.

Solution: Calculation was correctly in the App.





New in Version v2.0.1 b (9)

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

1) Display Action/Tasks in the Footprint

In order to quickly assess system state and debug configuration issues the users need to see the EPP Tasks and their state at a glance.

Requirement: Display the Tasks Configured in a Footprint along with their:

- TYPE (TT, PP, SC, SN, etc.)
- SLKS Number (For quick reference to hardware)
- SLKS State (For quick state determination)

Action / Task that is executing...



Action / Task that is waiting...



Action / Task is the not required...



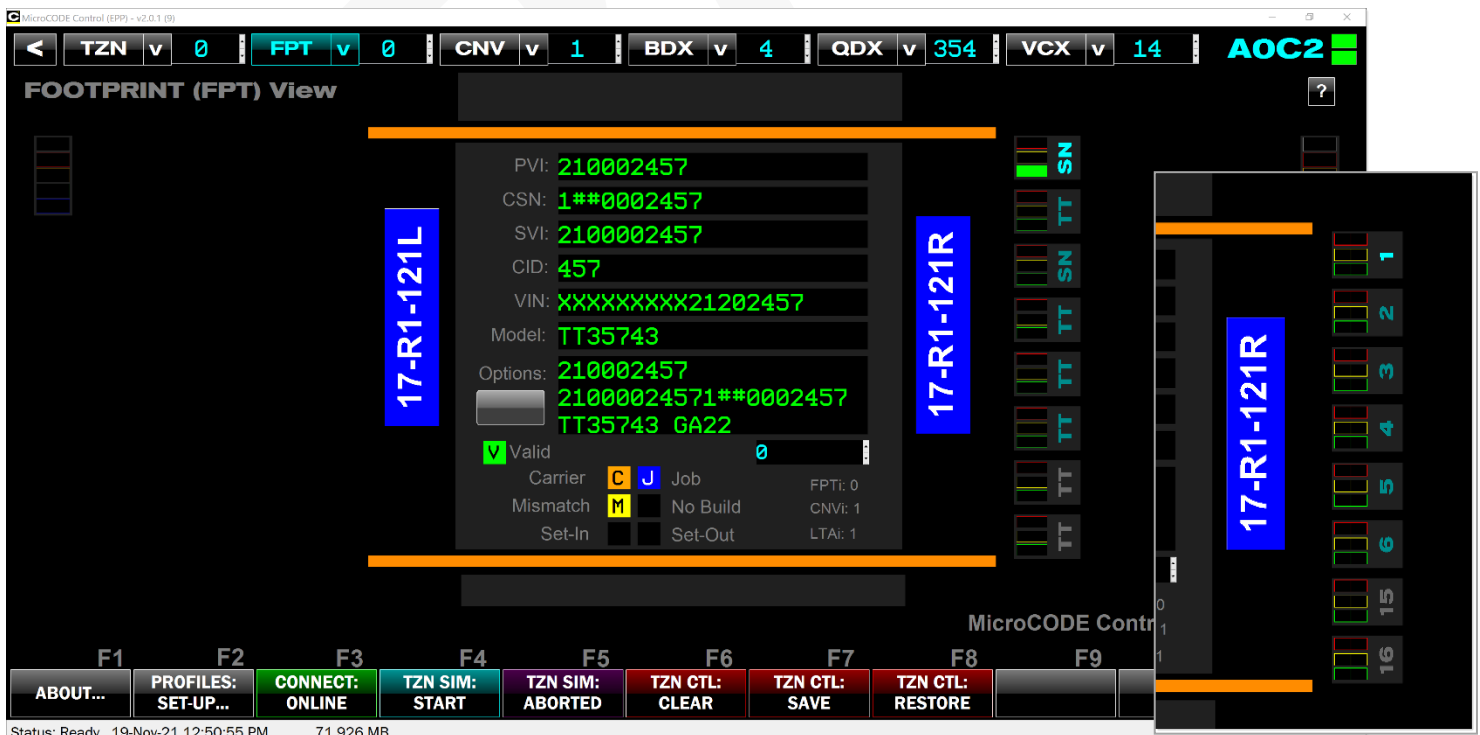
Implementation: The Control App now displays up to sixty-four (64) Tasks in the Footprint. Up to thirty-two (32) on each side.

Only the Tasks configured in the Footprint are displayed.

For each Task the App shows a representation of the LED lighted Key Switch (SLKS), along with its state, the SLKS Number (#), and the acronym representing the Task TYPE. (The TYPE and SLKS alternate in the display).

TYPEs:

- TT** = Torque Tool
- PP** = Part Pick
- BC** = Part Scan (Barcode Scan)
- SN** = Part Sensor(s)
- TR** = Verify Component (Trace Scan)
- PT** = Process Tool
- VS** = Verification Station
- CC** = Custom Code
- VV** = Vision Verification





Implementation (Cont.):

Clicking on any of the display Action's open a detailed view into the Configuration and Status of the Task...

Example: Task that is executing...

TASK Status

SN

GSF1

TASK: 12640

X

GEO SET FIXTURE 1

SLKS: 1

PVI: 210002457

CSN: 1##0002457

17-R1-121L

Task Status Viewer

<p>Job</p> <p><input checked="" type="checkbox"/> Job Present <input type="checkbox"/> Job Mismatch <input type="checkbox"/> No Build Job</p> <p><input type="checkbox"/> Prereq. Complete <input type="checkbox"/> Past Start FP <input type="checkbox"/> Bad Build Data</p> <p><input type="checkbox"/> Work Started <input type="checkbox"/> Work in Progress <input checked="" type="checkbox"/> Work Enable</p> <p><input type="checkbox"/> Pre Warning FP <input type="checkbox"/> Past Warning FP <input type="checkbox"/> Work Complete</p> <p><input type="checkbox"/> Pre Stopped FP <input type="checkbox"/> Past Stopped FP <input type="checkbox"/> Warning Point (>PWP)</p>	<p>Order</p> <p><input checked="" type="checkbox"/> Work Required</p> <p><input checked="" type="checkbox"/> Action Required</p>	<p>U.I.</p> <p><input type="checkbox"/> Enable (Command)</p> <p><input type="checkbox"/> Bypass (Command)</p> <p><input type="checkbox"/> Release (Command)</p> <p><input type="checkbox"/> In Progress (Command)</p> <p><input type="checkbox"/> Retrigger Request</p> <p><input type="checkbox"/> Work Disabled</p>
---	---	--

Part Sensor(s)

OK

Cancel

Example: Task that is waiting for 'ordered execution'...

TASK Status

TT

SG DROP REG SEC

TASK: 12644

X

[SSB-1103-04] - SWING GATE DROP F

SLKS: 15

PVI: 210002457

CSN: 1##0002457

17-R1-121R

Task Status Viewer

<p>Job</p> <p><input checked="" type="checkbox"/> Job Present <input type="checkbox"/> Job Mismatch <input type="checkbox"/> No Build Job</p> <p><input type="checkbox"/> Prereq. Complete <input type="checkbox"/> Past Start FP <input type="checkbox"/> Bad Build Data</p> <p><input type="checkbox"/> Work Started <input type="checkbox"/> Work in Progress <input type="checkbox"/> Work Enable</p> <p><input type="checkbox"/> Pre Warning FP <input type="checkbox"/> Past Warning FP <input type="checkbox"/> Work Complete</p> <p><input type="checkbox"/> Pre Stopped FP <input type="checkbox"/> Past Stopped FP <input type="checkbox"/> Warning Point (>PWP)</p>	<p>Order</p> <p><input type="checkbox"/> Work Required</p> <p><input type="checkbox"/> Action Required</p>	<p>U.I.</p> <p><input type="checkbox"/> Enable (Command)</p> <p><input type="checkbox"/> Bypass (Command)</p> <p><input type="checkbox"/> Release (Command)</p> <p><input type="checkbox"/> In Progress (Command)</p> <p><input type="checkbox"/> Retrigger Request</p> <p><input type="checkbox"/> Work Disabled</p>
--	---	--

Torque Tool

OK

Cancel





Example: Task that has exceeded its 'work envelope', and left in 'unknown state' after a 'data shift'...

TASK
Status

TT

TG STKR MIN SEC

[SSB-1116-03] - TAIL GATE STRIKER

TASK: 11345

SLKS: 9

X

PVI: 210002458 CSN: 1##0002458 17-R1-122L

Task Status Viewer

Job	<input checked="" type="checkbox"/> Job Present	<input type="checkbox"/> Job Mismatch	<input type="checkbox"/> No Build Job	Order	<input checked="" type="checkbox"/> Work Required
	<input type="checkbox"/> Prereq. Complete	<input type="checkbox"/> Past Start FP	<input type="checkbox"/> Bad Build Data		<input type="checkbox"/> Action Required
Work	<input type="checkbox"/> Work Started	<input type="checkbox"/> Work in Progress	<input type="checkbox"/> Work Enable	U.I.	<input type="checkbox"/> Enable (Command)
	<input type="checkbox"/> Pre Warning FP	<input type="checkbox"/> Past Warning FP	<input type="checkbox"/> Work Complete		<input type="checkbox"/> Bypass (Command)
	<input type="checkbox"/> Pre Stopped FP	<input type="checkbox"/> Past Stopped FP	<input type="checkbox"/> Warning Point (>PWP)		<input type="checkbox"/> Release (Command)
O.I.	<input type="checkbox"/> Bypassed	<input type="checkbox"/> Bypass (Key)	<input checked="" type="checkbox"/> Stopped (@FPS)	<input type="checkbox"/> In Progress (Command)	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; background-color: red; color: white; font-weight: bold; font-size: 1.2em;">R</div> <div style="border: 1px solid black; padding: 5px; background-color: yellow; margin-bottom: 5px; width: 20px; height: 20px; margin: 0 auto;"></div> <div style="border: 1px solid black; padding: 5px; background-color: green; color: white; font-weight: bold; font-size: 1.2em;">G</div>
	<input type="checkbox"/> Released	<input type="checkbox"/> Release (Key)	<input checked="" type="checkbox"/> Work Failed	<input type="checkbox"/> Retrigger Request	
I/O	<input type="checkbox"/> Released Required	<input type="checkbox"/> Communication Error		<input type="checkbox"/> Work Disabled	
	<input type="checkbox"/> Faulted	<input type="checkbox"/> Out-of-Tethering			
	<input type="checkbox"/> Battery Low				

Torque Tool

OK Cancel

DRAFT



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.



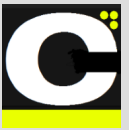
6) ALS/AGV [E] / [L] Commands don't work first time (SEP/EPP)

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.



New in Version v2.0.1 b (10)

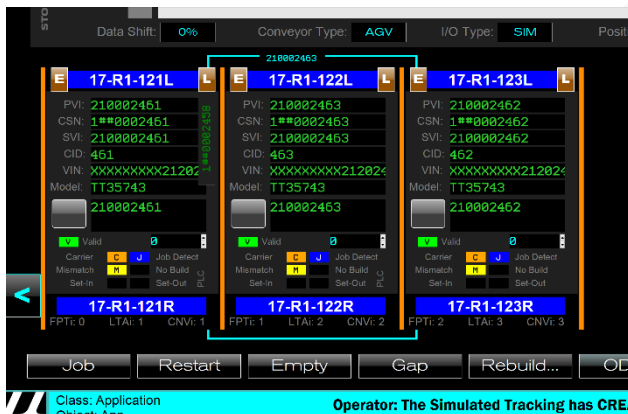
The following features were added to **MicroCODE Control (EPP)** in this Release:

1) CSN Sequence Generation in a Single Footprint, better CSN control

The App can now generate a complete CSN Sequence for testing all in a single Footprint to facilitate Stop Station testing among other things.

Requirement: In previous version the App always created the next highest CSN in a Track Zone when a new Job was generated. This is fine on a running FPS Conveyor, but when testing Stop Stations is inconvenient when you want to run thru a sequence of Job conditions.

Example: Clicking “Job” in the position shown below would always clear CSN 463 and just recreated it (463 be the next CSN in the Track Zone after it is erased.



Implementation: Not the App is aware of the Job it is replacing and increments the CSN—and hence JOB PROFILE—every time you click “Job” on the same Footprint.



NOTE: The REBUILD command was also updated to start with the CSN in Footprint instead of 1.

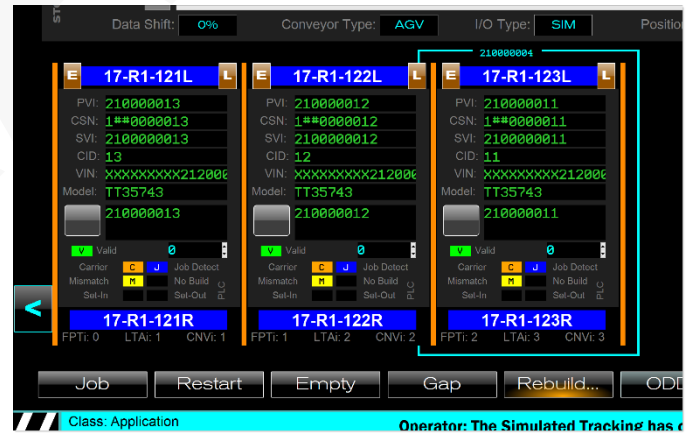
A blank Track Zone will always REBUILD starting at CSN Zero (0).

Starting with CSN 007, you can click “Job” to increment the CSN in that Footprint as many times as required...

Each click of “Job” will created the next CSN in Footprint:



Then you can click “Rebuild” and the entire Track Zone will fill in from that point...



NOTE: To reset your CSN Sequence to ZERO clear the Track Zone and start over.





Corrected in Version v2.0.1 b (10)

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

1) GEPICS Formats not accessible

User selected GEPICS Formats were not accessible after selection preventing App use.

Issue: Build v2.0.1b9 was failing to copy the user selected Formats into their private App directory during the selection process.

Solution: This was corrected to properly copy all selected Formats (Area and Zone) into the User's private App Data folder during selection. (This allow selection off removable media—like USB Sticks—and later use of the App without the media preset).

Quick Work-around (in lieu of updating to Build 10:

Manually copy all your AREA formats to:

C:\Users\<<Username>\Documents\MicroCODE Control (EPP)\GEPICS Formats\Area

Manually copy all your ZONE formats to:

C:\Users\<<Username>\Documents\MicroCODE Control (EPP)\GEPICS Formats\Zone

And select them from there.

2) PROFILE Edits are not always saved

Users make changes to Job Profiles and the changes disappear, sometimes.

Issue: In all prior releases the App stop saving Jobs and Profile Changes at the first unconfigured Row. So, the Build Data modification shown below on Row 9 would be lost because Rows 5 – 8 were 'unconfigured'. This was done for speed and to minimize the size of saved CFG Files. But, it was not obvious at all, and the App did not warn the user that these edits would be lost.

CSN #	Change	HEADER	Position	Length	Data	OPTIONS	Position	Length	Data
1	PVI	0	9	16	NNNNNNNN3	PVI	0	9	NNNNNNNN3
2	SVI	16	10	26	NNNNNNNN3	SVI	16	10	NNNNNNNN3
3	CSN	26	11	54	NNNNNNNN3	CSN	26	11	NNNNNNNN3
4	WIN	54	17	38	NNNNNNNN3	WIN	54	17	NNNNNNNN3
5	MODEL	38	8	n/a	MM0000	MODEL	38	4	MM03
6	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
7	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
8	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
9	NONE	n/a	n/a	n/a	n/a	RPO	333	3	UM3
10	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
11	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
12	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a

Solution: The App now saves all rows—including 'unconfigured' rows—up to and thru the last row holding instructions for modifying the GEPICS Order.

All the way to [99] the last Row index allowed (100 different changes per CSN #).

CSN #	Change	HEADER	Position	Length	Data	OPTIONS	Position	Length	Data
88	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
89	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
90	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
91	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
92	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
93	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
94	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
95	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
96	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
97	NONE	n/a	n/a	n/a	n/a	NONE	n/a	n/a	n/a
98	NONE	n/a	n/a	n/a	n/a	RPO	900	3	UM9
99	NONE	n/a	n/a	n/a	n/a	RPO	901	3	UMX





Corrected in Version v2.0.1 b (11)

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

1) JOB Buffer Tags are not always shown

After a User pushes a Job out of a Stop Station it 'disappears' and the Vertical Tag that normally shows the presence of the Job in the PLC's Buffer Tags is missing from the display.

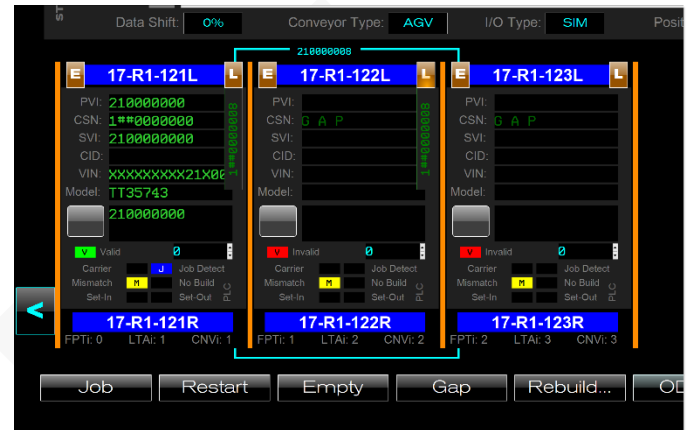
Issue: The internal 'Carrier Present' for the Buffer Footprints was 'false' even though a Job was present in the Buffer, this 'Carrier Present' flag is what determines if the Job position is displayed in the vertical tag.

Vertical Buffer Tag is missing:



Solution: 1) The App now assumes that any Job has a 'Carrier' under regardless of the PLC status of that flag, and 2) The App now auto-focuses on newly create Jobs during simulation commands issued by the user.

Normal Display (also with v2.0.1b10):



Corrected displayed with v2.0.1b11 for all cases:





2) FOCUS is not updated on NEW JOBS

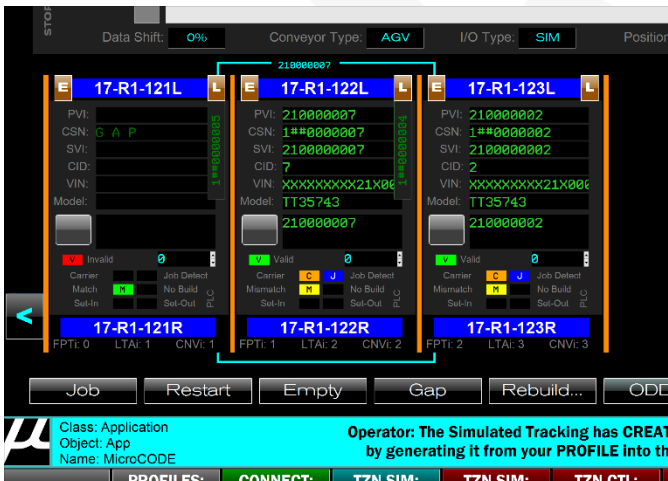
After a User creates a new Job in Footprint the 'JOB FOCUS' is not updated automatically to that PVI.

Issue: The App is not auto-focusing on the created PVI after it appears following a user command like "Job" to create a new PVI.

Immediately following a 'Job' creation command the JOB FOCUS is still gray with no PVI in the wire-frame. This indicates the App is not 'following that PVI.



Solution: The App now auto-focuses on newly create Jobs during simulation commands issued by the user. (Indicated by the PVI in the cyan wire-frame).





Corrected in Version v2.0.1 b (12)

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

1) Next CSN Generates DUPLICATE

When creating new Jobs, the App keeps making CSN 0000 every time.

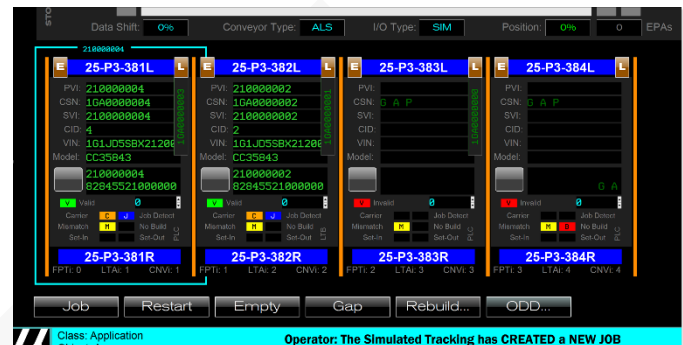
Issue: The App reads the Tracking Zone to determine what the next CSN should be. After scanning the Track Zone, it takes the highest CSN it finds, increments it and uses that for the next Job.

The only exception is using the JOB command on top of another which will increment the CSN in the Footprint regardless of the other CSNs in the Track Zone. (This can product a duplicate).

The root cause here was the App not scanning the 'inter-Footprint Buffer spaces' in the EPP PLC. (Those shown as Vertical Labels in ALS/AGV/VAC Lines).

Solution: The App now scans both the Tracking Image and Tracking Buffer to determine the highest CSN in the Track Zone.

Corrected displayed with v2.0.1b12:



2) The JOB FOCUS was not following conveyor movement

After focusing on a Job with the JOB FOCUS (Cyan Outline) the focusing was not following the Job as it moves.

Issue: After the User clicks on a Footprint with a Job the PVI takes the 'JOB FOCUS' and that focus will follow the PVI as it moves and pans the screen to keep it visible.

This was temporarily broken by Correction #2 in Build (11).

Solution: The App now distinguished between a NEW JOB being creates and the movement of an EXISTING JOB.





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

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

No new features

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

The following defects were fixed in **MicroCODE Control (EPP)** 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.





New in Version v2.0.2 b (3)

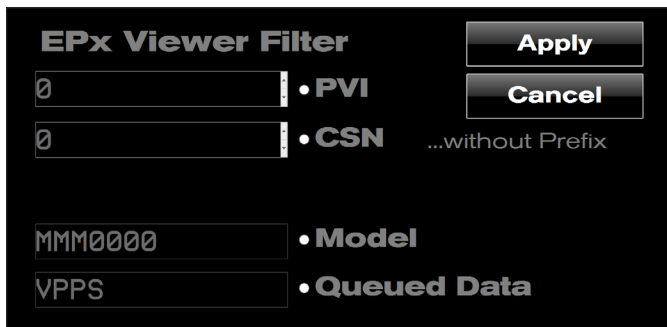
The following features were added in this Release:

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 EPP PLC API needed documentation corrections for proper installation

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

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

Correction: Correct placement of the MicroCODE API being in simulation mode was 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)

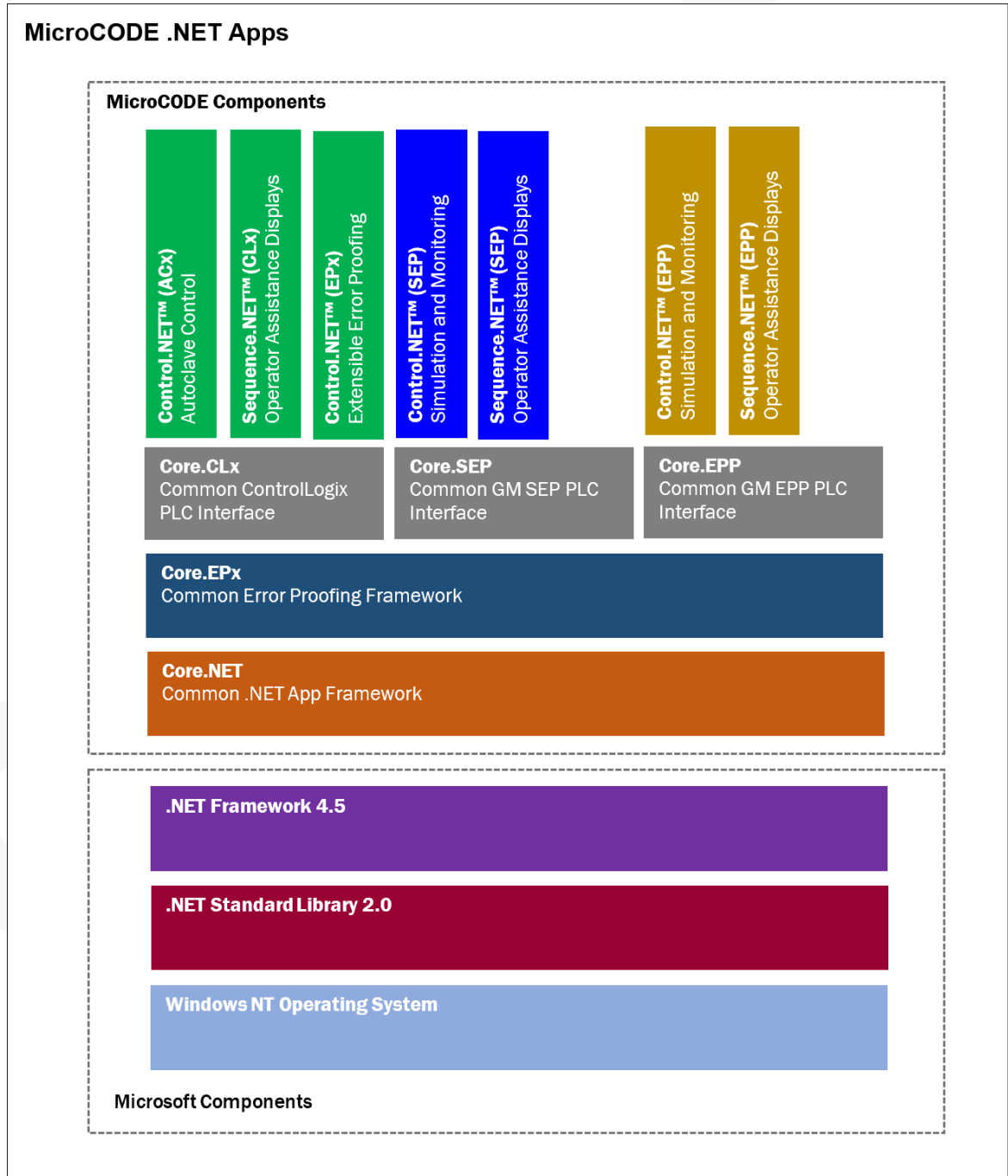
The following features were added in this Release:

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.**

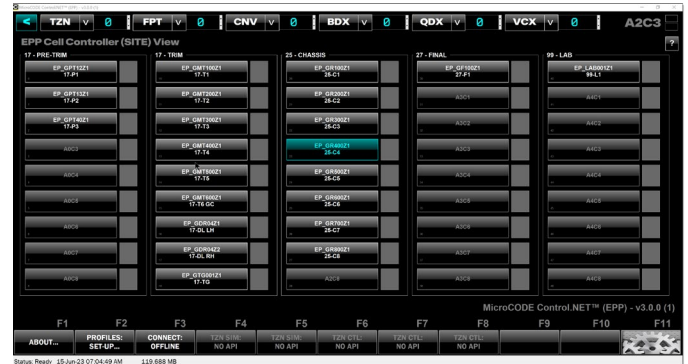
Test Case 9: Resize/Minimize/Maximize the App on a FHD monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

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.**

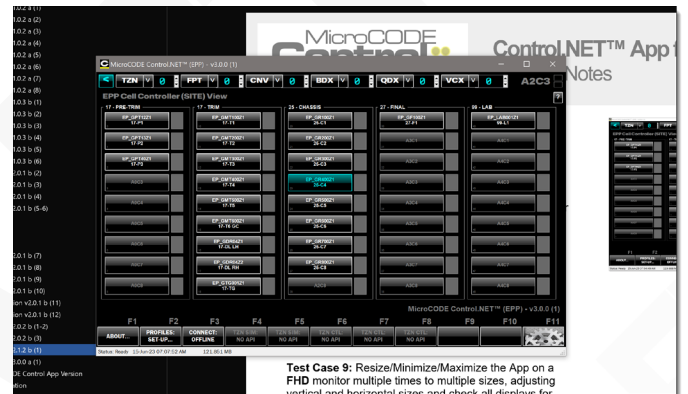
Test Case 11: Resize/Minimize/Maximize the App on a XGA monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

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....



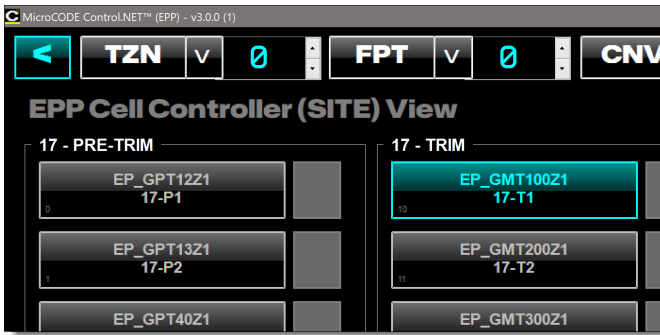
Scaling

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.**



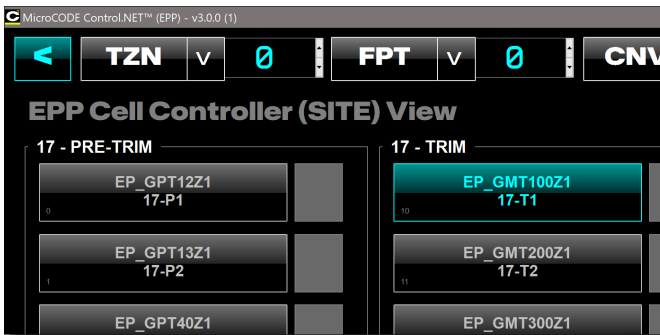


Without our 'Resizer':



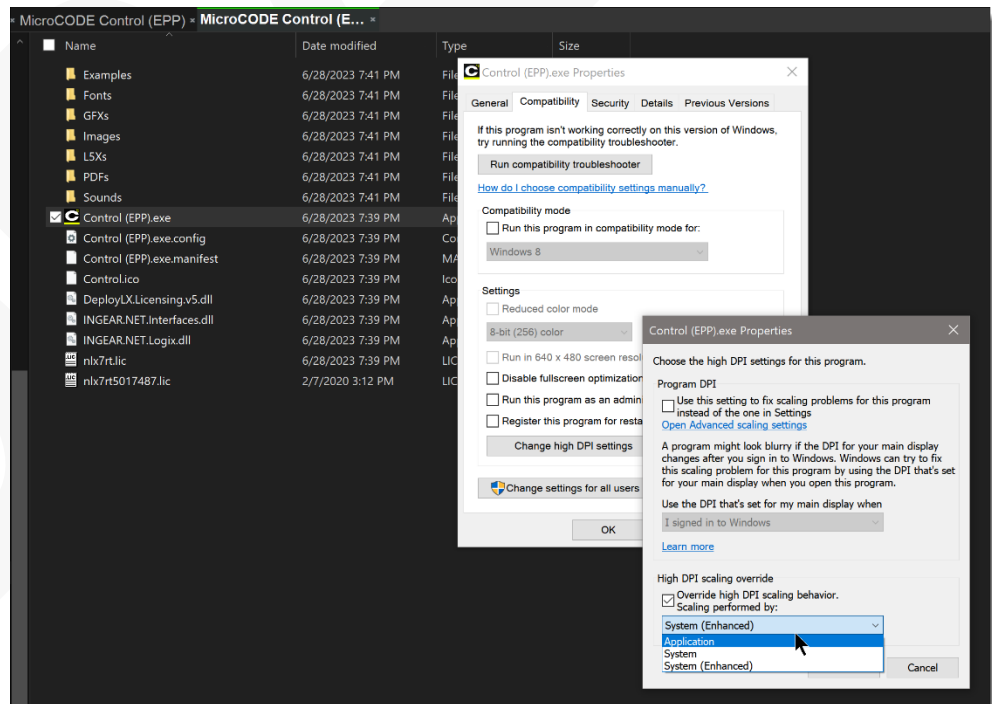
This update is critically dependent on a proper 'App Manifest' being delivered during installation. This is now included (along with an 'App Config' file). The 'Manifest' ensures that Windows 10 knows that the App handles all resizing, scaling, DPI changes, and support 'High DPI' (4K) monitors.

With our 'Resizer':



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'.

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? Uggh.





Control.NET™ App for GM EPP

Release Notes Alpha Release: 3.0.0 a (1)



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"
3 <description>Control.NET™ (EPP)</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 <dpiAwareness> 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
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

```

```

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"
3 <description>Control.NET™ (EPP)</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 <dpiAwareness> 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
46 <dpiAwareness xmlns="http://schemas.microsoft.com/SM/2016/WindowsSettings" PerMonitorV2 />
47
48 <!-- Legacy (pre-Windows 10 v1507)
49
50 <dpiAwareness> 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 <!-- A list of the Windows versions that this application has been tested on
66 <!-- and is designed to work with. Uncomment the appropriate elements
67 <!-- and Windows will automatically select the most compatible environment. -->
68
69 <!-- Windows Vista -->
70 <!--supportedOS Id="{e281457f-1546-4355-d5fe-888d5e339819}" />-->
71
72 <!-- Windows 7 - *removed* from supported list to stop 'Compatibility Mode' -->
73 <!--supportedOS Id="{95966641-6811-691b-8026-04484227959a}" />-->
74
75 <!-- Windows 8 - *not* supported -->
76 <!--supportedOS Id="{4a228228-2269-4441-b99c-d99994039898}" />-->
77
78 <!-- Windows 8.1 *not* supported -->
79 <!--supportedOS Id="{1f976c70-b0e1-4239-90bb-838b7600a787}" />-->
80
81 <!-- Windows 10 - supported by this MicroCODE App -->
82 <supportedOS Id="{8e777121-bf72-4f6b-b565-48788b55a478}" />
83
84 </application>
85 </compatibility>
86
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.

The screenshot shows the application's main window with the following text:

MicroCODE Control.NET™ (EPP)
 Version 3.0.0 Alpha (1) - Major.Minor.Release Cycle (Build)

Machine: LENOVO 20QNCT01WW
 OS: Microsoft Windows NT 10.0.19045.0
 .NET: 4.0.30319.42000 (.NET Full Framework)

Copyright © 2012-2023 MicroCODE, Inc.
 Designed & Coded by: MicroCODE Incorporated

A dialog box is open showing the SIL License for MONA-SANS GitHub Font, Copyright © 2017, Deni Anggara. All rights reserved. The dialog also includes the SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007 and a PREAMBLE section. The dialog has 'OK' and 'Cancel' buttons.



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\EP_GRCSTZ1\
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 (EPP)..."
40 }
```

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.

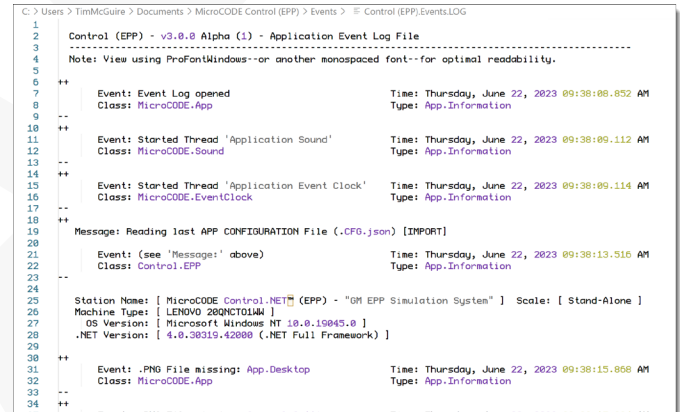
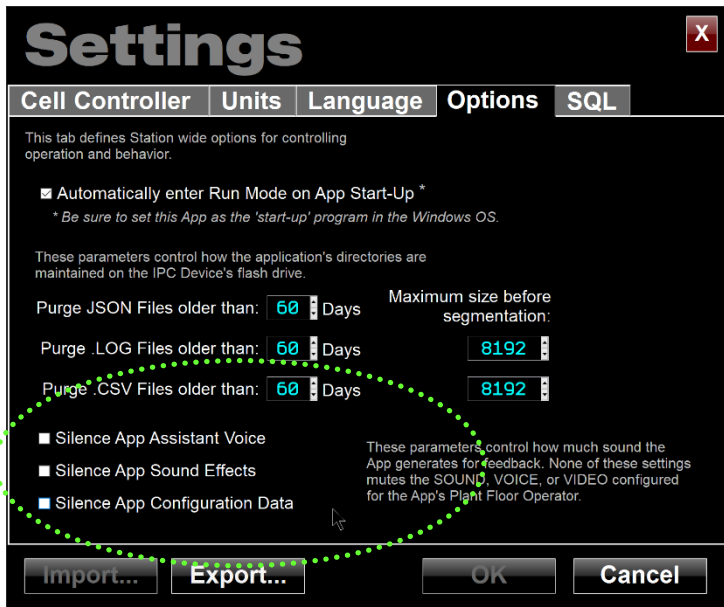
This .LOG file was not intended for Developer messaging, debug statements, data logging.

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

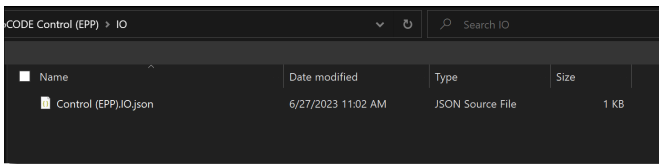
The Control.NET™ Apps now support persistent I/O memory via JSON file storage.

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) EPP Database Hierarchy Support

The Control.NET™ Apps were originally designed to support single station Autoclaves, then SEP, and then EPP. The Autoclave and SEP implementations supported their fixed layouts, i.e.: one (1) Station, or thirty (30) Cell Controllers in ten (10) Panels (Areas).

Requirement: EPP does not have a 'fixed' configuration hierarchy, but instead is free-form and completely left up to the user to configure on a plant-by-plant basis.

Implementation: The main App Screen is now free-form and re-designs itself based on the EPP SQL database layout.

The current internal limit is (10) Areas x (10) Cells each, (100) EPP PLCs, but even this can be changed in seconds if needed in the future with three constants...

```

/// <summary> This is the APP's PERMANENT MEMORY, i.e.: THE STORAGE OF CONFIGURA ...
/// </summary>
public class Settings
{
    // PUBLIC CONSTANTS
    #region PublicConstants

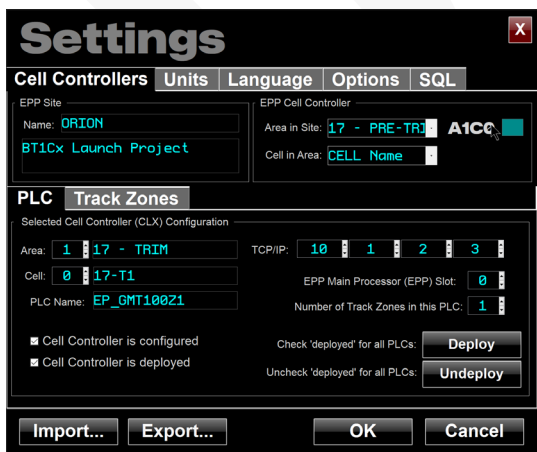
    /// <summary>
    /// The Maximum number of EPP SQL Areas processed during import.
    /// </summary>
    public const int MaximumSqlAreas = 99;

    /// <summary>
    /// The Maximum number of EPP Areas (arbitrary) supported by this App.
    /// This is used as the default until an EPP SQL DB is queried for the actual number.
    /// The actual number is taken to be the number of Areas with defined EPP PLCs.
    /// </summary>
    public const int MaximumAppAreas = 10;

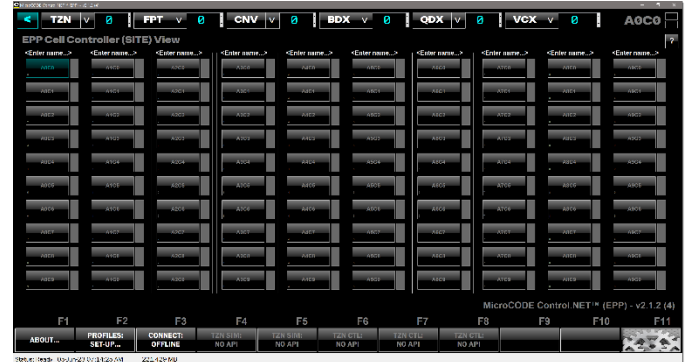
    /// <summary>
    /// The Maximum number of EPP Controllers in an Area (arbitrary) supported by this App.
    /// This is used as the default until an EPP SQL DB is queried for the actual number.
    /// The actual number is taken to be the largest number of EPP PLCs within an Area.
    /// </summary>
    public const int MaximumAppCells = 10;
    }
    
```

Now when the Control.NET App queries the EPP SQL Database it processes each Shop (BS, PS, GA...) and only import Areas (Departments 17, 19, 25...) with defined EPP Cells Controllers.

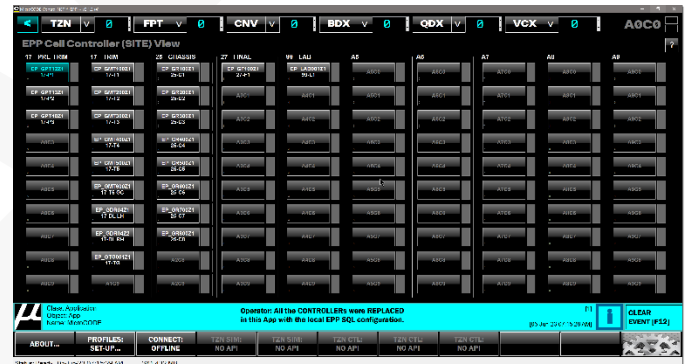
The SETTINGS Dialog was redesigned to accommodate this change as well...



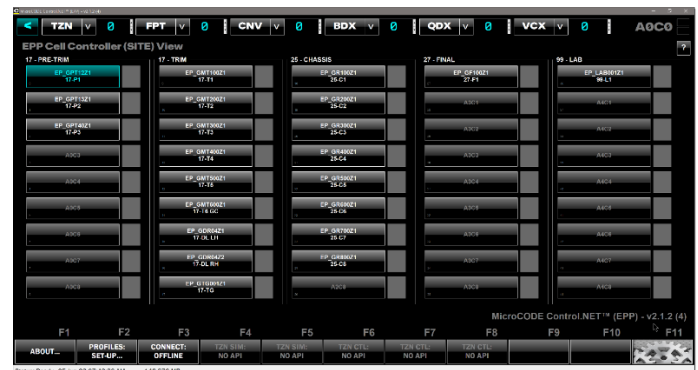
Blank App displays its default 10 x 10 (100) EPP Cell Controllers...



Immediately after a query of the EPP SQL database the App shows the Areas and EPP PLCs that are configured...



All subsequent uses of the App with this configuration will display a 're-designed' front-end that conforms to the Site configuration...





9) Generate defaults GEPICS Formats and GEPICS Seed Jobs

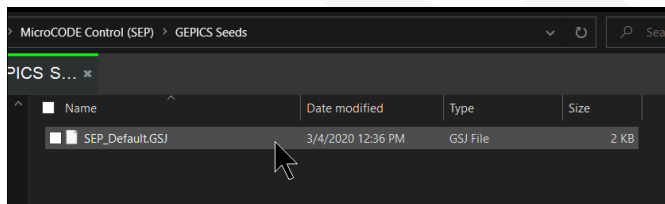
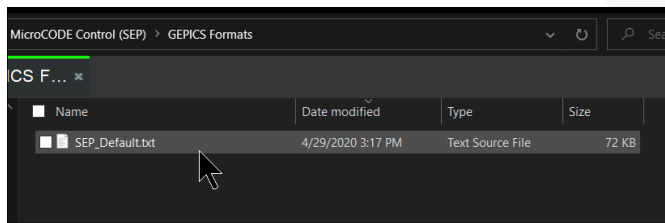
The Control App needs to see the GEPICS Format(s) (both AREA and TRACK ZONE for EPP) when interacting with the EPP or SEP PLCs.

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.



10) 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

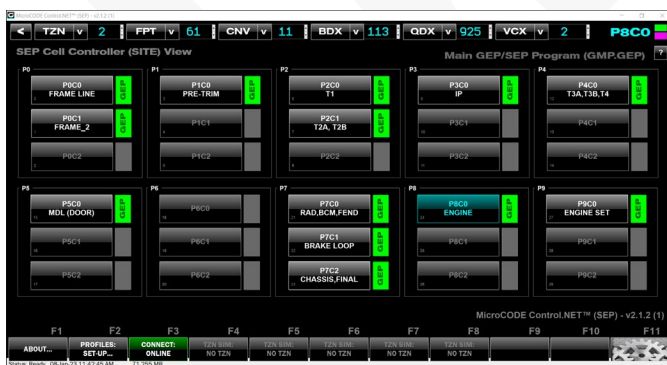
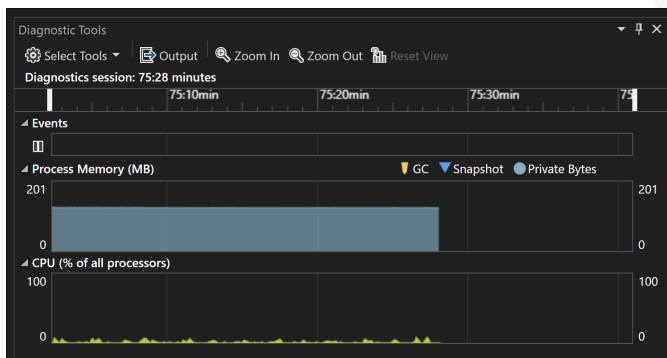
The App is now 10x faster than previous versions and uses less than 7% of available CPU time.

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) Incompatibility with EPP v1.418+

The App was not working with the new ORION (BT1Cx) FPS Conveyor, no simulated movement was happening.

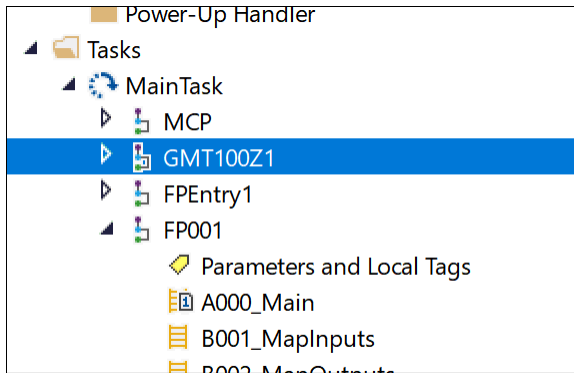
Issue: Fixed Position Stop (FPS) Conveyor simulation was no longer working in v2.1.x.

Correction: The App was taking the EPP SQL Track Zone name as the name of the Logix 5000 Program Name. This is not always true, it was for Factory Zone, it that is not enforced by the EPP UI, nor should it be. The App was corrected to use the EPP PLC Program Name to generate the names of the Track Zones programs.

Note: There is also a difference at ORION (BT1Cx), where the “TZn” in the EPP Names has been shortened to just “Zn”. The App handles either case.

PLC: EP_<CNVNAME>TZ1
 EP_<CNVNAME>Z1

TZ Program: <CNVNAME>TZ1-n
 <CNVNAME>Z1-n



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

Test Case 1: Test Simulation with Factory Zero SQL DB and PLC Programs. – **Result:** ...

Test Case 2: Test Simulation with Orion SQL DB and PLC Programs. – **Passed:** All Cells, Track Zones, and Footprints are handled properly.

4) 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.





5) Support Deleted Objects in EPP SQL

Previous versions of the App were seemingly bring in objects—Track Zones, Tasks, etc.—into the wrong Areas of an Epp configuration.

Issue: The EPP UI does not actually delete object from its SQL Database when they are deleted by a User. Some of the queries in this Control.NET app did not take that into account, e.g.: TRIM 1, TRIM 2, and TRIM 3 were deleted from the PRE-TRIM Area in the ORION SQL DB, but they are still in there, marked “-1” in their “PositionIndex” and “3” in their “ItemStateld”.

Correction: All EPP SQL queries were updated to filter out the deleted objects.

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

Test Case 1: Using the existing ORION SQL DB re-query the entire configuration into the Control.NET (EPP) App. – **Passed:** None of the deleted objects are present and empty Areas are removed from view in the App.

Test Case 2: Using the existing FACTORY ZERO SQL DB re-query the entire configuration into the Control.NET (EPP) App. – **Result:** Test pending at time of v3.0.0a1 release...

Id	Name	Description	PositionIndex	NodeIdentifier	ParentNodeId	NodeTypeId	PlantId	ItemStateld	ExecutionTarg
1	1020	17-P1	0	NULL	1008	3	29	1	1
2	1021	17-P1 GC	1	NULL	1008	3	29	1	1
3	1022	17-P2	2	NULL	1008	3	29	1	2
4	1023	17-P2 EB	3	NULL	1008	3	29	1	2
5	1024	17-P3	4	NULL	1008	3	29	1	3
6	1025	17-P3 TH	5	NULL	1008	3	29	1	3
7	1026	17-T1	-1	NULL	1008	3	29	3	4
8	1027	17-T2	-1	NULL	1008	3	29	3	5
9	1028	17-T3	-1	NULL	1008	3	29	3	6
10	1029	17-T1	0	NULL	1009	3	29	1	4
11	1030	17-T2	1	NULL	1009	3	29	1	5
12	1031	17-T3	2	NULL	1009	3	29	1	6
13	1032	17-T4	3	NULL	1009	3	29	1	7
14	1033	17-T5	4	NULL	1009	3	29	1	8
15	1034	17-T6 GC	5	NULL	1009	3	29	1	9
16	1035	17-T6	6	NULL	1009	3	29	1	9
17	1036	17-DL LH	7	NULL	1009	3	29	1	10
18	1037	17-DL RH	8	NULL	1009	3	29	1	11
19	1038	17-TG	9	NULL	1009	3	29	1	12
20	1039	25-C1	0	NULL	1010	3	29	1	13
21	1040	25-C1 RT	1	NULL	1010	3	29	1	13
22	1041	25-C2	2	NULL	1010	3	29	1	14
23	1042	25-C3	3	NULL	1010	3	29	1	15
24	1043	25-C4	4	NULL	1010	3	29	1	16
25	1044	25-C5	5	NULL	1010	3	29	1	17





Control App Software Version Numbers

The 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 **'Minor 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 (1)
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 application.

MicroCODE Control App – Build Matrix				
Version	Win NT OS	.NET	Logix 5000	EPP CDE + FW
v1.m.R	Windows 7 or 10	4.72	Up to L8x	v1.0.0 + v32.xx
v2.m.R	Windows 7 or 10	4.8	Up to L8x	v1.417 + v32.xx
v3.m.R	Windows 10	4.5	Up to L8x	V1.418 + 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.

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

