

Plex Supply  
Chain Community



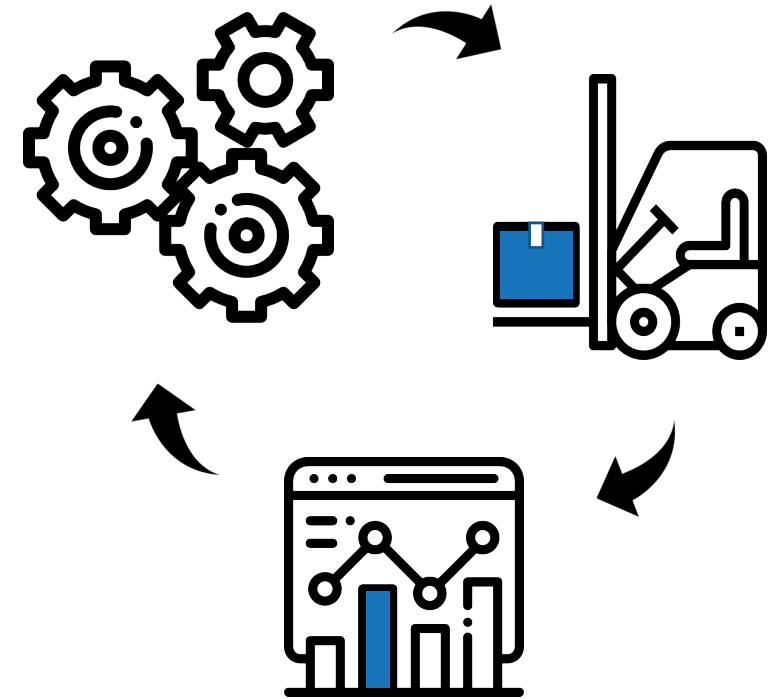
# Automated Line Replenishment

Luke Abrams, Newman Technology



# Why?

- We were experiencing significant downtime waiting for parts.
  - Material Handlers and Machine Operators were disconnected.
  - Material Handlers were driving around looking for what to do next.
  - We had zero insight into our Material Handler activity/performance.
    - Whose fault was it that the lines were going down?
- So, we made our machines automatically send the requests.
  - Increasing labor utilization.
  - Creating data and analytics on our material handling process.
  - Increasing safety on our shop floor.
  - Delivering ROI by reducing downtime waiting for parts.
- How did we get here?



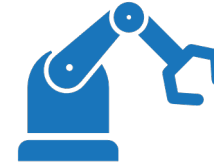
# Newman Technology, Inc.



Japanese Automotive  
Supplier



4 Manufacturing Facilities in  
North America



Automated Manufacturing  
Process

- Focused on automation and machine integration
  - 25% Plex Control panel
  - 75% Kors/Mach2
- Smaller dedicated resources for material handling.
  - Leveraging Cella Shop Mobile for scanning.
    - Keyence & Windows Mobile Devices (going away soon)
  - Dedicated Material Handlers assigned to specific lines.

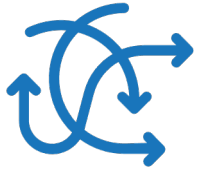


# The Problem “waiting for parts”



Manufacturing came to a halt because material wasn't in the right place at the right time.

---



Materials Team was doing their best, but they were always reacting. VP of Operations was sick of watching forks driving empty “looking for work”.

---



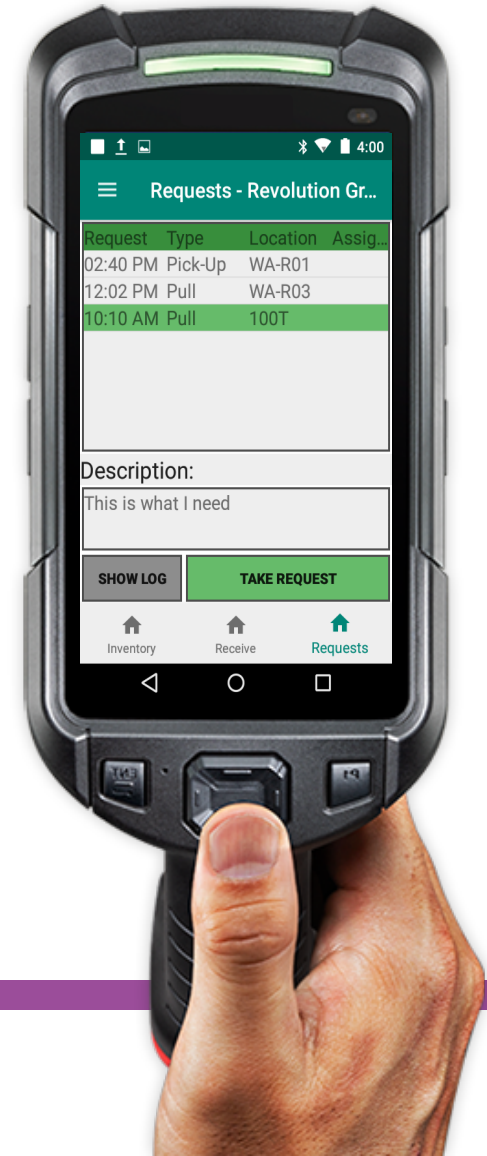
Needed to find a way to connect our Material Handlers into the process and provide modern tools to do their job.



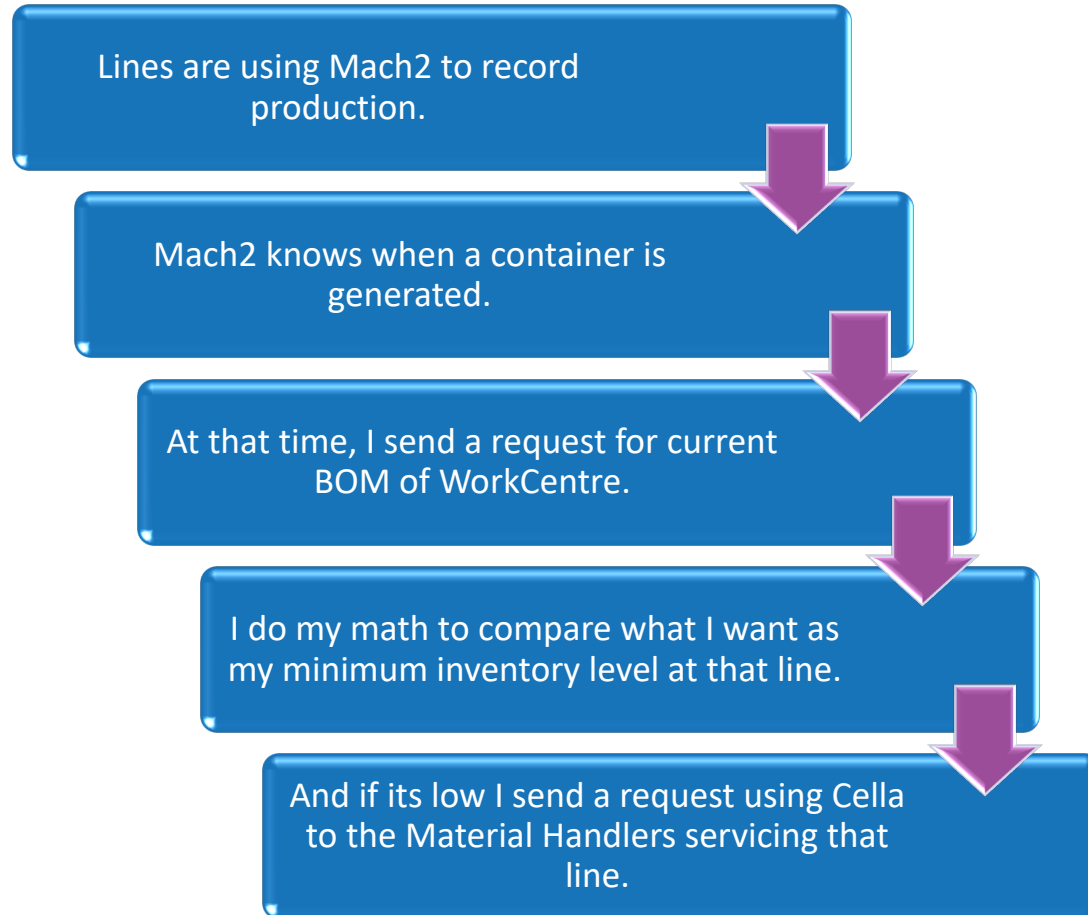


# Partnering with Cella Technologies

- Cella Request Management
  - New API Layer
- Aligning with Kors Engineering
- All 3 teams worked together to deliver the vision of machines automatically making requests to material handler.



# How Does it Work?



The below are the API requests for Cella Request Management. The returned data is parsed to the right.

The logic below is taking the request retrieved from Cella, parsing the based on shift schedules setting that get the request. On 1st shift and comps are 2 separate groups on 2nd shift it is one group that gets

**CellaURL**  
String Writable  
Out: `https://api.cella-tech.com/api/v1/RequestManagement`

**ResourceWorkcenter**  
Dynamic Links  
FlexRecordSuccess: false (ok) @ 16  
PLCName: 1120TVA10 (ok) @ 16  
WCKey: 54773 (ok) @ 16  
CurrentShift: 1st A - 5:00AM (ok) @ 16  
PartNumber: 67351TVA A010Y1-11Y1 (ok) @ 16

**ReplaceString**  
Replace String  
In String: `https://api.cella-tech.com/api/v1/RequestManagement`  
In String To Replace: Workcenter (ok)  
In Replacement String: 1120TVA10 (ok) @ 16  
Out String: `https://api.cella-tech.com/api/v1/RequestManagement`

**ResourceGroupGet**  
Simple Rest Client  
URI: `https://api.cella-tech.com/api/v1/RequestManagement/ResourceGroup/locations/1120TVA10`  
Method: Get  
Login:  
Password:  
Body:  
Response: `{("resourceGroupID":42,"resourceGroup": "Sash Comp"), ("resourceGroupID":43,"resourceGroup": "Sash Child")}` (ok)  
Message: 11/06/2020 15:05:55 Success  
Transaction Time: 0.00  
Send

**ResourceGroupParse**  
String Splitter Dynamic  
In String: `{("resourceGroupID":42,"resourceGroup": "Sash Comp"), ("resourceGroupID":43,"resourceGroup": "Sash Child")}` (ok)  
In Delimiter: , (ok)  
Out Selected Item: `{("resourceGroupID":42,"resourceGroup": "Sash Comp")}` (ok)  
Out Items: 6 (ok)  
outField\_00: `{("resourceGroupID":42,"resourceGroup": "Sash Comp")}` (ok)  
outField\_01: "resourceGroup": "Sash Comp" (ok)  
outField\_02: "resourceGroupID":43 (ok)  
outField\_03: "resourceGroup": "Sash Child" (ok)  
outField\_04: "resourceGroupID":47 (ok)  
outField\_05: "resourceGroup": "2nd Shift Sash" (ok)

**ResourceGroupParse1**  
String Splitter Dynamic  
In String: "resourceGroup": "Sash Comp" (ok)  
In Delimiter: , (ok)  
outField\_00: (ok)  
outField\_01: resourceGroup (ok)  
outField\_02: (ok)  
outField\_03: Sash Comp (ok)  
outField\_04: (ok)  
outField\_05: (ok)

**ResourceGroupParse3**  
String Splitter Dynamic  
In String: "resourceGroup": "Sash Child" (ok)  
In Delimiter: , (ok)  
outField\_00: (ok)  
outField\_01: resourceGroup (ok)  
outField\_02: (ok)  
outField\_03: Sash Child (ok)  
outField\_04: (ok)  
outField\_05: (ok)

**ResourceGroupParse4**  
String Splitter Dynamic  
In String: "resourceGroup": "2nd Shift Sash" (ok)  
In Delimiter: , (ok)  
outField\_00: (ok)  
outField\_01: resourceGroup (ok)  
outField\_02: (ok)  
outField\_03: 2nd Shift Sash (ok)  
outField\_04: (ok)  
outField\_05: (ok)

**1stShiftCheck**  
String Match  
Check  
In A: 1st A - 5:00AM (ok) @ 16  
In B: 1st (ok)  
Out Match: true (ok)  
Out No Match: false (ok)  
Out String: 1st A - 5:00AM (ok)  
Match

**2ndShiftCheck**  
String Match  
Check  
In A: 1st A - 5:00AM (ok) @ 16  
In B: 2nd (ok)  
Out Match: false (ok)  
Out No Match: true (ok)  
Out String: (ok)  
Match

The number set below is how many containers worth of child/source parts is the minimum desired to have lineside.

**NumberOfContainersEarly**  
When Numeric Setpoint  
Out: 3.00 (ok)  
Changed By: (ok)  
Time Changed: null

The string update folder below replaces parts of the string that gets sent via API to Cella based on the request type. For full containers of comps it sends the comp number. For child parts/source inventory it sends component part and the part number.

**StringUpdate**  
Function Folder  
TriggerFull: false (ok) @ 16  
SetPartName: Component Part 6735F-TVA FR PLR ASSY (ok) @ def  
PartID: 67351TVA A010Y1 (ok)  
FullContainerPartNumber: 67351TVA A010Y1 (ok)

The Plex Inventory folder below checks the source inventory required by the BOM from Plex and checks the currently loaded quantity to the workcenter. It compares to a quantity determined by how many containers early and if the source is lower than that minimum number will trigger a request for the child/source part.

**PlexInventoryLevel**  
Function Folder

**Or**  
Out: false (ok)  
In A: false (ok) @ 16  
In B: false (ok)

**SendRequest**  
Fire On Change  
Boolean In: false (ok)  
New Boolean Input

**RequestManagementSend**  
Simple Rest Client  
URI: `https://api.cella-tech.com/api/v1/RequestManagement/ResourceGroup/Request`  
Method: Post  
Login:  
Password:  
Body: `{("resourceGroup": "Sash Child", "location": "1120TVA10", "description": "6735F-TVA FR PLR ASSY", "requestType": "Pull", "itemNo": "6735F-TVA FR PLR ASSY", "containerCount": 1), ("requestID": 191934, "succeeded": true)}` (ok)  
Response: `{("requestID": 191934, "succeeded": true)}` (ok)  
Message: 02/02/2021 13:36:08 Success  
Transaction Time: 606.00  
Send

# Successes



Creating a culture change for the forklift operators.

- Hesitant at first but came around after using it and realizing how it makes their job easier.



People are looking at what they keep lineside.

- Is it too much, too little?
- Is what Plex says we have at the line accurate?



Insight into who the high performers are.



Reducing downtime waiting for parts!





# Next Steps- Min/Max

- User friendly automated requests based on inventory levels set in Cella.



Q&A

