**EPOS**

Documents contains the current implementation logic for EPOS.

**Application status**

We are using the following application status.

RED **- 0**

WARNING **- 5**

GREEN **- 10**

OTHER **- 15 (**in the case of*File doesn’t exist* or *File is empty* etc.**)**

This is the general scenarios we gets while reading the file**.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **File** | **File status** | | | | |
|  | File doesn’t exist | File is empty | File is corrupted | File contains invalid data  ( based on general conditions ) | File is valid and applying field specific logic. |
| **A**. PED Audit Metrics And Threshold | [**A****1**](#A1) | [**A****2**](#A2) | [**A****3**](#A3) | [**A****4**](#A4) | [**A****5**](#A5) |
| **B.** Controller Memory | [**B****1**](#B1) | [**B****2**](#B2) | **[B3](#B3)** | [**B****4**](#B4) | [**B5**](#B5) |
| **C.** Store General Information | [**C****1**](#C1) | [**C****2**](#C2) | [**C****3**](#C3) | [**C****4**](#C4) | **[C5](#C5)** |
| **D.** Critical File health status | [**D****1**](#D1) | **[D2](#D2)** | [**D****3**](#D3) | [**D****4**](#D4) | [**D****5**](#D5) |
| **E.** BGCHECK | **E1** | **E2** | **E3** | **E4** | **E5** |
| **F.** DECCHECK | **F1** | **F2** | **F3** | **F4** | **F5** |

**[A1](#A1p):** - Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

[**A****2**](#A2p)**:** - Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[**A****3**](#A3p)**:** - Sends “***File is corrupted one***” message to dashboard.  
 Application status is set to **15**.

[**A4**](#A4p)**:** - Applying general condition and storing all invalid rows in a temporary array.

[**A****5**](#A5p)**:** - Reads temporary array contains the invalid rows and Checks below condition and update application status.

**Green condition:** More than 98% of the tills in the estate has got a working PED

**Amber condition:** Number of tills with PED serial number in between 95% to 98%

**Red condition:** If more than 5% is down in the estate then red condition

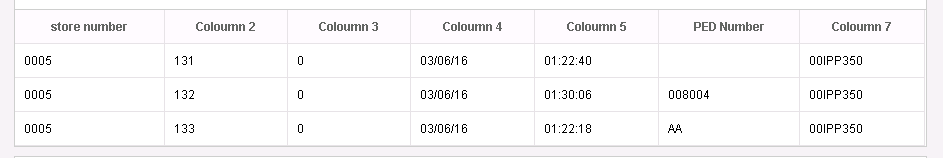
In **Green** condition we are not displaying the invalid PED details in dashboard. Just showing the message as below.

XXXX stores and XX tills have been checked and all Tills looks fine.

OR

XXX stores and XX tills have been checked and XX number of tills have incorrect PED numbers.

In **Amber** and **Red** condition displays all the invalid PED details.



[B1](#B1p):- Currently doesn’t provide any details to dashboard.

[B2](#B2p):- Currently doesn’t provide any details to dashboard.

[B3](#B3p):- Currently doesn’t provide any details to dashboard.

[B4](#B4p):- Skips all the invalid rows.

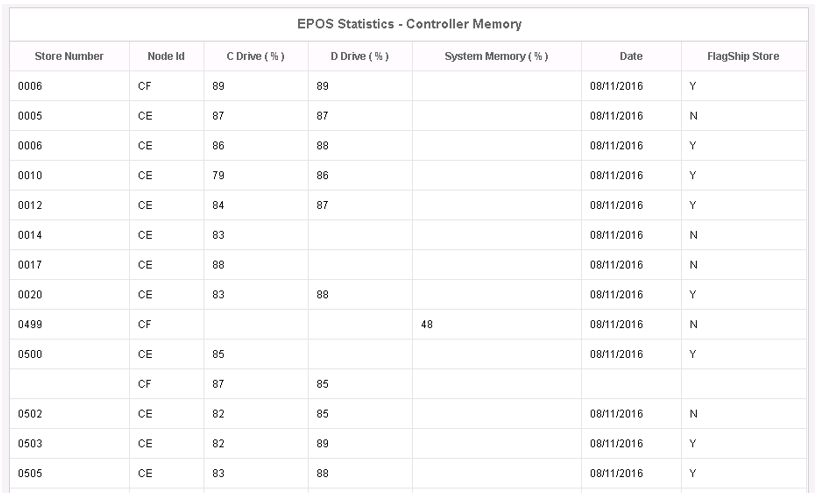
[B5](#B5p):- Here checks the following conditions

* Divide **C drive free memory** by **C drive total memory**, if the value is **less than 90%** then move the message in to an error array.
* Divide **D drive free memory** by **D drive total memory**. If the value is **less than 90%** then move the message in to an error array.
* **Divide free memory** by **total memory** and if the free memory available is **lesser than 50%** then move the message in to an error array.

**Amber** - Number of Stores (other than flagship) which fail this condition checks < 10

**Red** - If any Flagship stores fails in the condition check OR Number of Stores (other than flagship) which fail this condition checks >= 10

*Dashboard view for this is as shown below,*



**Green :** No more issue in fields. Dashboard message is as shown below.  


[C1](#C1p):- Not implemented

[C2](#C2p):- Not implemented

[C3](#C3p):- Not implemented

[C4](#C4p):- Not implemented

[C5](#C5p):- Not implemented

[D1](#D1p):- Currently doesn’t provide any details to dashboard.

[D2](#D2p):- Currently doesn’t provide any details to dashboard.

[D3](#D3p):- Currently doesn’t provide any details to dashboard.

[D4](#D4p):- Currently skips the row.

[D5](#D5p):- The following way we are reading and sending the status.

"**Case A**"

"Divide 5th field by 4th field (used Recs/Total Recs), if the result is more than 85 % it’s a red alert for that store.

There are 24 files being checked against each store. So even if its single file is in issue, raise a red alert with store and file details". (Copied from PPT)

"**Case B**"

While reading the file, if we gets a row has longest chain value greater than 6.So we are ready to send "**amber**" status.

"**Case C**"

We have 24 files for each store.

We are grouping the store and finding the longest chain.

The count of 'longest chain value greater than 6' is greater than 12 for each set of store, we are ready to send "**red**" status.

Otherwise we moves to next set of stores and finds the count of 'longest chain value greater than 6'.

"**Case D**"

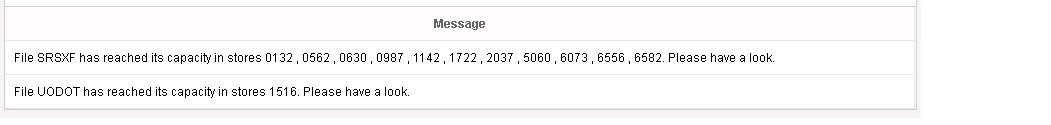
If “**case A**”, “**case B**” and “**case C**” doesn’t exists.

At end of file we sends "**Red**" status if “**case A**" OR "**case C**" exists.

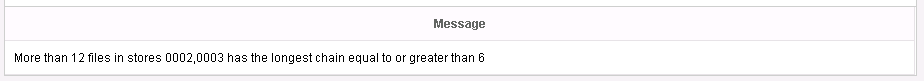
At end of file we sends "**Amber**" status if “**case B**" exists.

At end of file we sends "**Green**" status if “**case D**" exists.

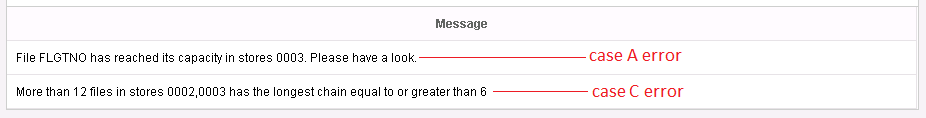
**“Case A” errors**



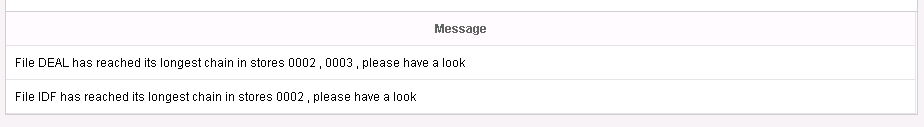
**“Case C” errors**



**“Case A” and “Case C” errors**

****

**“Case B” errors**

****

**“Case D”**

Dashboard shows the message as shown below.

XXX stores have been checked. 24 files from each store has been checked for empty space and chain factors and all looks fine.