**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](#PED_B) | [**A****1**](#A1) | [**A****2**](#A2) | [**A****3**](#A3) | [**A****4**](#A4) | [**A****5**](#A5) |
| **[B.](#CMEM_B)** [Controller Memory](#CMEM_B) | [**B****1**](#B1) | [**B****2**](#B2) | **[B3](#B3)** | [**B****4**](#B4) | [**B5**](#B5) |
| **[C.](#SGI_B)** [Store General Information](#SGI_B) | [**C****1**](#C1) | [**C****2**](#C2) | [**C****3**](#C3) | [**C****4**](#C4) | **[C5](#C5)** |
| **[D.](#CFHS_B)** [Critical File health status](#CFHS_B) | [**D****1**](#D1) | **[D2](#D2)** | [**D****3**](#D3) | [**D****4**](#D4) | [**D****5**](#D5) |
| [**E.**BGCHECK](#BGCK_B) | [**E****1**](#E1) | [**E****2**](#E2) | [**E****3**](#E3) | [**E****4**](#E4) | [**E****5**](#E5) |
| [**F.**DECCHECK](#DECK_B) | [**F****1**](#F1) | **[F2](#F2)** | [**F****3**](#F3) | [**F****4**](#F4) | [**F****5**](#F5) |

**[PED Audit Metrics and Threshold](#PED)**

**Frequency of check** : Twice a day.  
**Task scheduled at** : 4.06 am  
 12.06 pm

[**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. (If any one of the rows is corrupted)  
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 :** More than 98% of the tills in the estate has got a working PED

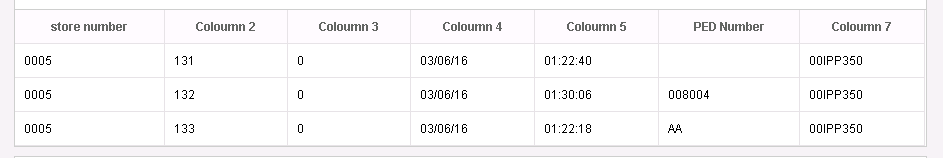
**Amber :** Number of tills with PED serial number in between 95% to 98%

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

If **100%** of the tills in the estate has got a working PED then it shows the message in information level as follows.

XXXX stores and XX tills have been checked and all tills have correct PED numbers.

In **Green** (if it is not 100% working PED), **Amber** and **Red** condition displays all the invalid PED details.



**[Controller Memory](#CMEM)**

**Frequency of check** : Once a day.  
**Task scheduled at** : 3.06 am

[B1](#B1p):- Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

[B2](#B2p):- Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[B3](#B3p):- If any corrupted row is present then it skips and reads next row.

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

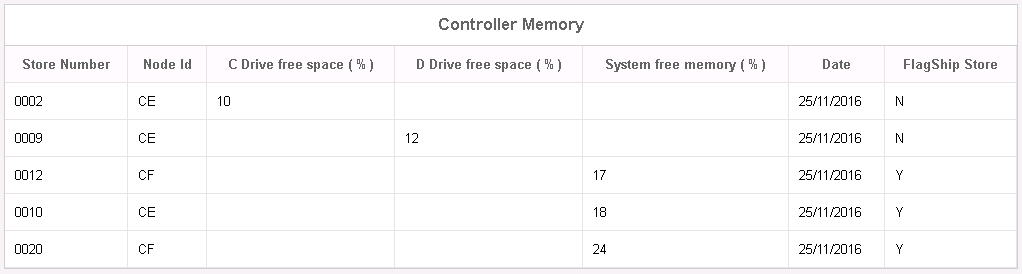
[B5](#B5p):- Here checks the following conditions

* Divide **C drive free memory** by **C drive total memory**, if the value is **less than 15%** 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 15%** then move the message in to an error array.
* **Divide free memory** by **total memory** and if the free memory available is **lesser than 25%** then move the message in to an error array.

**Amber : Number of Stores which fail this condition checks > 25**

**Red** : **Number of Stores which fail this condition checks >50**

*Dashboard view for this is as shown below, ascending order of system free memory*



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


[**Store General Information**](#SGI) **Currently not implemented any logic and not integrated to dashboard.**

**Frequency of check** : Once a day.  
**Task scheduled at** : 12.06 pm

[C1](#C1p):- Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

[C2](#C2p):- Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[C3](#C3p):- Skips the corrupted row.

[C4](#C4p):- There is no need to check any general conditions.

[C5](#C5p):- Showing all the records to the dashboard. There is no need to update application status based on any aggregate conditions.

**[Critical File health status](#CFHS)**

**Frequency of check** : Second of Every month.  
**Task scheduled at** : 3rd of every month 2.36am

[D1](#D1p):- Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

[D2](#D2p):- Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[D3](#D3p):- Skips the corrupted row.

[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.

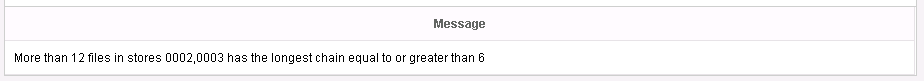
At end of file we sends "**Amber**" status if “**case B**" exists.

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

**“Case A” errors**



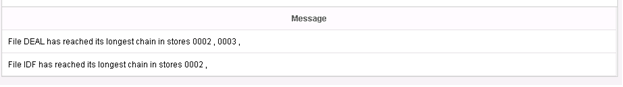
**“Case C” errors**



**“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.

**Don’t check the report now as it’s not correct**

**[Background application check](#BGCK)**

**Frequency of check** : 9 times a day.  
**Task scheduled at** : 09.25 am  
 10.25 am  
 11.25 am  
 12.25 pm   
 13.25 pm   
 14.25 pm   
 15.25 pm   
 16.25 pm   
 17.25 pm

[E1](#E1p):- Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

[E2](#E2p):- Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[E3](#E3p):- Not applicable.

[E4](#E4p):- Not applicable.

[E5](#E5p):- For BGCHECK we need to look for the keyword STORE and fetch the 4bytes towards the right. Data will be in below format

*One or more critical background applications are not active in the following stores:*

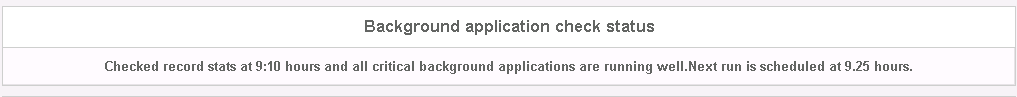
***STORE****5604*

***STORE****0002*

The dashboard message looks like as

**Red :** **If number of stores failed are greater than 50**  


**Amber :** **Number of stores affected are greater than 25**  
  
**Green :** If there are no store number is present in the file. The message will display as,



**[DEC Tickets](#DECK)**

**Frequency of check** : 6 times a day.  
**Task scheduled at** : 09.06 am  
 11.06 am  
 13.06 pm  
 15.06 pm   
 17.06 pm   
 20.06 pm

[F1](#F1p):- Sends “***file doesn’t exist message***” message to dashboard.  
 Application status is set to **15**.

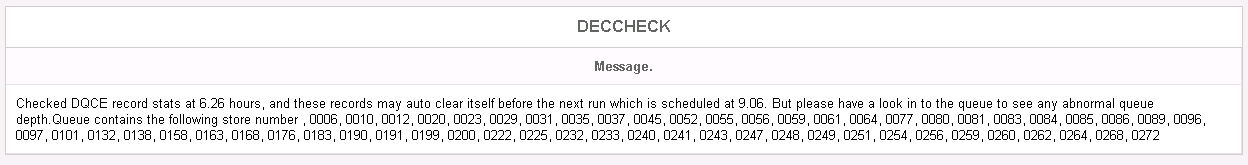
[F2](#F2p):- Sends “***File contains no information***” message to dashboard.  
 Application status is set to **15**.

[F3](#F3p):- Not applicable.

[F4](#F4p):- Not applicable.

[F5](#F5p):- There are two queues we monitor to make sure data flows fine without issues between systems.  
 **a. DQCE  
 b. DQ2CE**

Any queue with unprocessed records count more than a limit (limit is 50) should be reported to dashboard as follows.



**Red :** **If more than *50* stores have issues with any queue**

**Amber :** **If stores have issues with any queue is in between *25 to 50***

**Green :** **up to 25 should be green**