Matrix Documentation

The Matrix

Shift Is Split into Partials	Has OFF Segment	Does Shift Duration = Paid Hours?	Working Segment is 5 Hours or Less (WFM will not deduct a break when this is true)	OFF Shift Duration = Paid Hours of OFF?	WFM Calculates	ESP Schedule Calculates (_ESCData)	ESP Timecard Calculates
No	N/A	N/A	N/A	N/A	Correct	Correct	
Yes	No	Yes	N/A	N/A	Correct	Wrong	
Yes	No	No	Yes	N/A	Correct	Wrong	
Yes	No	No	No	N/A	Correct	Correct	
Yes	Yes	No	Yes	Yes	Wrong	Wrong	Correct
Yes	Yes	No	Yes	No	Correct	Maybe (depends on the length of	Maybe - More likely to be right
Yes	Yes	No	No	No	Wrong	Wrong	Wrong

Objective of the Matrix

This matrix leverages five key factors related to shift types within a pay period to determine whether each shift is processed correctly by ESP (the legacy system), WFM (the future-state system), or both.

When applied to a large dataset, such as an entire pay period, the matrix provides valuable insights:

- It enables a direct comparison of system accuracy and performance, helping quantify the improvements offered by WFM over ESP.
- It allows for targeted filtering of cases where WFM is expected to produce errors, facilitating deeper analysis and troubleshooting.

How the Matrix is Applied to the Dataset

Each row in the dataset represents a shift worked during the pay period. These shifts are evaluated against five predefined conditions (outlined in the factors list below). Based on how each shift aligns with these conditions, we assign an outcome indicating whether it will be handled correctly by WFM, ESP, or both systems.

This mapping allows us to:

- Predict which shifts are likely to encounter issues in WFM
- Identify discrepancies between systems for further analysis and validation

Limitations of the Matrix:

- This matrix is designed exclusively for straight time shifts, defined as regular shifts that exclude any overtime (OT) extensions.
- As a result, it cannot be used to analyze OT shifts. A separate matrix would need to be developed to evaluate those cases.
- The dashboard applying this matrix has filtered the PP14 dataset to exclude all OT extensions, ensuring alignment with the matrix's intended scope.
- Additionally, if WFM configurations have been updated since the matrix was created, its predictive accuracy may be affected. Changes to system rules or logic could alter how shifts align with the matrix's factors, potentially leading to incorrect outcomes.

Potential Integration with Machine Learning

- The integration of this matrix with a machine learning (ML) model is intended to significantly reduce the time spent on manual payroll comparisons. For example, by applying the matrix to a dataset of 500,000 rows, we could narrow the scope to approximately 200,000 rows -focusing only on those shifts where WFM is likely to produce inaccuracies. This allows us to confidently exclude rows that are expected to be correct, streamlining the analysis process.

- However, regular collaboration with the data team is strongly recommended to ensure this approach aligns with their intended objectives and to validate that the matrix continues to support evolving analytical needs.

Dashboard Data Transformation Guide

- **Is Shift is Split into Partials?**
  - Dataset has a column called Partials populated with 0s and 1s.
  - If 1, the shift is a partial shift
  - If 0, the shift is a non-partial shift

- **Has OFF Segment**
  - Dataset has a column called PayCodeType
  - If the value in this column is 2, it is an off shift
  - If the value in this column is anything else, it is not an off shift

- **Does Shift Duration = Paid Hours? **Requires Time-Based Adjacency Logic**
  - Dataset has the two columns: Shift Start Time & Shift End Time
  - These are date time fields
  - If you find the difference of these two, it will give you the duration of the shift
  - There are other columns in the dataset that allude to some sort of working hours, however, data team they can be unreliable so this is the best way to go
  - Use this logic to apply the correct break to hours of the shift to be able to compare it to paid hours:

```
= Table.AddColumn(#"Rename PartialShiftFlag -> IsPartialShift", "DurationEqualsPaid", each if
[DurationHours] <= 5 then
if Number.Round([PaidHours], 2) = Number.Round([DurationHours], 2) then "Yes" else "No" else
if
[DurationHours] <= 8 then if Number.Round([PaidHours], 2) = Number.Round([DurationHours] -
0.5, 2) then "Yes" else "No"
else
if Number.Round([PaidHours], 2) = Number.Round([DurationHours] - 1, 2) then "Yes" else "No")
```

  - You will need to use some margin of error to round as due to arithmetic, the paid hours and duration hours might be off by 0.0001 or some arbitrary decimal value. These would still be considered equal to one another
  - Once you have a value for hours worked (or duration hours as I named it in the dashboard) compare it to the column PaidHours to see if they are equal or not

- **Working Segment is 5 Hours or Less**
  - Dataset has the two columns: Shift Start Time & Shift End Time
  - These are date time fields
  - If you find the difference of these two, it will give you the duration of the shift
  - There are other columns in the dataset that allude to some sort of working hours, however, data team they can be unreliable, so this is the best way to go
  - If the duration of the shift is 5 hours or less, can you Boolean mapping (1 = yes, is 5 hrs or less; 0 = no, it is not 5 hrs or less)

- **Off Shift Duration = Paid Hours of OFF? **Requires Time-Based Adjacency Logic**
    - Do the same transformation logic used in Shift Duration = Paid Hours, except we are not also adding another condition: is it an off shift?
    - So check the flag created for whether it is an off shift and then apply the shift duration = hours paid logic

****Time-Based Adjacency Logic**

A partial shift refers to a segment of a larger, continuous working block for an employee. For example, an employee may work from 07:00 to 19:00, with the day broken down as follows:

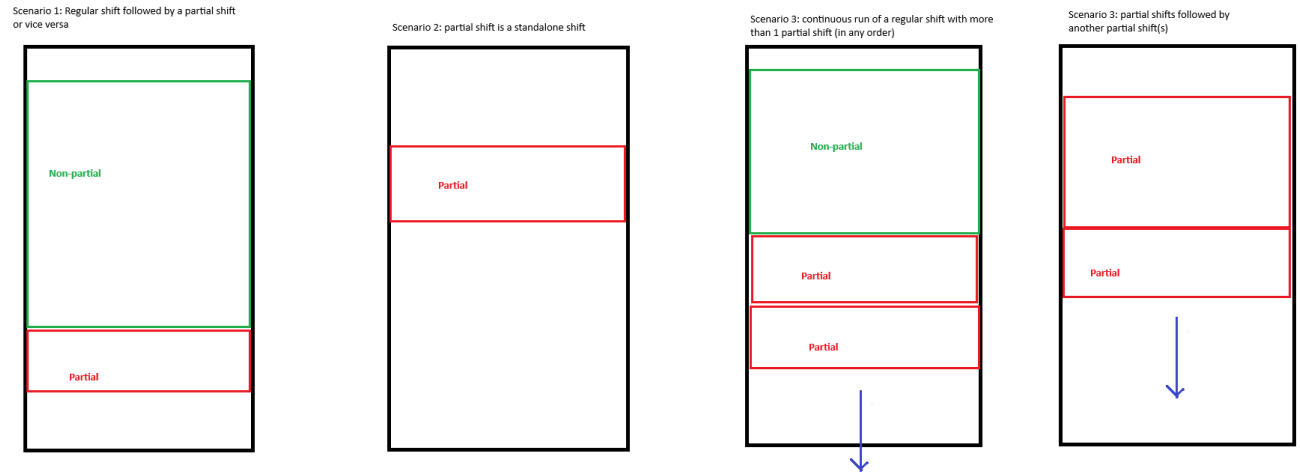
- 07:00–10:00: Medical Appointment
- 10:00–13:00: Regular shift
- 13:00–19:00: Relief shift

Although this is a single continuous workday, the dataset will represent it as three separate shifts, each associated with a distinct paycode (e.g., medical appointment, regular, relief). To determine whether **Shift Duration equals Paid Hours**, we must:

- Group adjacent shifts that occur back-to-back within the same working block
- Sum their total duration and corresponding paid hours
- Compare the aggregated values to assess alignment

Implementing this logic in the dashboard required three table merges to correctly group and evaluate adjacent shifts, making the process relatively complex. I refer to this methodology as **time-based adjacency logic**, since the key criterion for aggregation is whether shifts are contiguous in time.

A diagram on the following page illustrates the various scenarios that can arise when applying this logic.



Time-Based Adjacency Logic