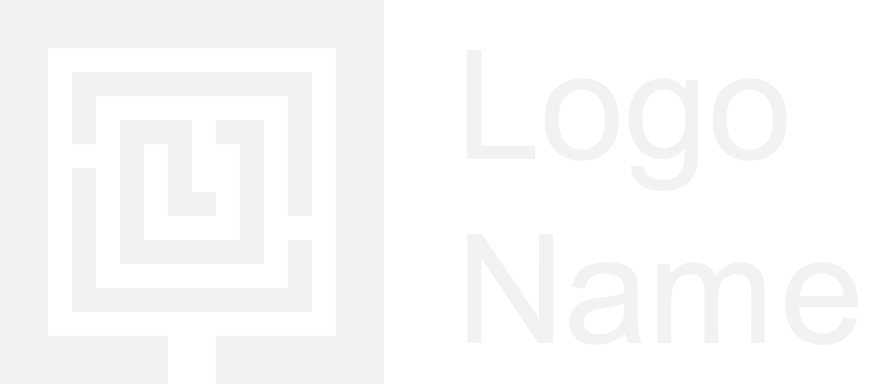


|  |
| --- |
| Validating The Orders Taken Report. |
|  |
| March 18    Authored by: Russell Kay |



# Validating The Orders Taken Report.

Contents:

1. Introduction
2. Validation Methods
3. Validation Issues and Resolutions
4. Report Analysis
5. Introduction

Welcome to the Orders Taken Report. All data used within this report is both Statistical and Quantitative. This report shows how many orders have been taken within the filtered time period. There are three time periods to choose from. 7 Days, Yesterday and Today.

Chart, treemap chart

Description automatically generated

You can also filter by Company, Cancelled Orders, Part Cancelled Orders and Not Cancelled Orders.

The report shows the number of orders taken, how many of these were cancelled and how many were part cancelled, as well as the values for each of these. The Report also shows which cancelation reasons were used.

The Data for the report is taken from the source system Dynamics Nav 2009. This system sits on top of SQL servers, a relational data base containing structured data stored in tables. The data from the source system is then transformed using SSIS packages and then loaded into the data warehouse, before being loaded into Azure and Displayed using Power BI.

We were asked to build this report by the sales directors who were looking to monitor the value and qty’s of orders received.

Below Is a screenshot of the Report.

Chart, pie chart

Description automatically generated

The Pie Charts Original Order Value by and Actual Order Value By, are broken down by customer. These Essentially show the value of orders by customer and the Value after cancelation by customer.

The top Right Corner of the report is showing the How many Items were ordered, The Value of the orders taken, The Margin made and what percentage of the sale was the margin. Each of those values have the actual amount before Cancelations Directly below them as well as the percentage difference between the two values.

The Unit Values and Unit Qty’s Pie charts are just visual representations of the figures in the top right.

The two remaining Pie Charts show the most used Cancelation reason codes based on Qty and Value. The last Pie chart shows the Value based on Housewares or electricals.

Now you are somewhat familiar with the report we will discuss how I validated the figures showing on the report. The Source System itself is an old version of Dynamics Nav from 2009.

In the next section we will look at the methods used.

1. Validation Methods

I created the code unit in the source system, Algorithms were used to decide if an order should be counted in the totals.

Before we look at the algorithm It is important to show you a section of the source data. Please see below.

Graphical user interface, application, table

Description automatically generated

This is only a small section of the table. There are many more columns that I just couldn’t fit with a single snapshot. However, you can see the Unit Price, Qty to be shipped and the sales order number. These are the three main columns we will be using.

As an example, the below screen shot shows an algorithm used in the Code unit on the source system.

Graphical user interface, text, application, email

Description automatically generated

This algorithm basically looks at order lines and decides whether the currency needs to be converted to GBP.

To determine if there was a currency factor (Exchange rate) I had to join the Sales Line Table to the Header Table in the Code. The Line of code to do this is shown below.

A picture containing diagram

Description automatically generated

The “.GET” basically allows us to ling the Header and the Lines by the 2 fields used within the brackets. The document Type and the Document Number.

The main block is an if statement. This first If essentially says if there is a sales line then do the following. The repeat causes this to repeat for every sales line that it finds until there is no more. “UNTIL SalesLine.NEXT() = 0;”

Then for each line that it finds it asks Does the Line contain VAT? If it doesn’t contain VAT, it moves on to ask if it has a currency factor, (This is the exchange rate if a currency other than GBP is used). If there is a currency factor, then the following calculation is used.

“T\_SalesLine."Unit Price" := SalesLine."Unit Price" / SalesHeader."Currency Factor"”

If there is no currency factor it does not need to use a calculation and simply takes the unit price from the order line, (T\_SalesLine.”Unit Price”).

“T\_SalesLine."Unit Price" := SalesLine."Unit Price"”

If, however there is VAT on the order then the same calculations are used as above but the results divided by 1.2. I.E.

“T\_SalesLine."Unit Price" := (SalesLine."Unit Price" / SalesHeader."Currency Factor") / 1.2”

This then Stores the Sales Line with the updated Unit Price in a temporary Table.” T\_SalesLine”

Because this would be an awful lot of data to go through, we also use filters to cleanse the data to make sure it is in the range we are looking.

The filters I used to cleanse the data were as follows.

1. “SalesLine.RESET();” This part resets the sales line to its default with no filters. So orders, Returns and Quotes are also shown.
2. “SalesLine.SETCURRENTKEY("Document Type",Type,"Sell-to Customer No.","No.",Quantity,"Qty Cancelled","Document Date");” This then sets the Key to make it faster to sort by organizing the data.
3. “SalesLine.SETRANGE("Document Type", SalesLine."Document Type"::Order);” Because quotes and returns share the same table, this filter makes sure that only sales orders are shown.

Before This filter is set Quotes and Returns Can be seen. Please see below.

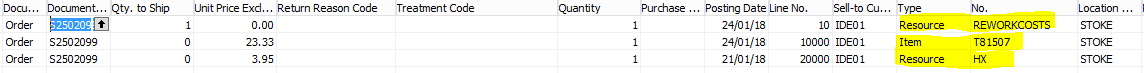
Table

Description automatically generated Table

Description automatically generated

1. SalesLine.SETRANGE(Type, SalesLine.Type::Item);” This filter then Makes sure the that the line is an item and not carriage or rework costs.

This image shows What the order lines of an order look like in the table if they have rework and carriage costs.

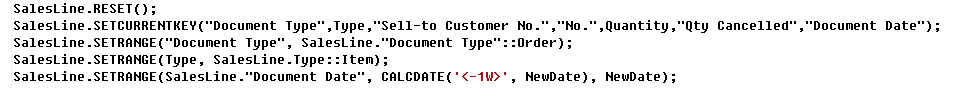


And then this is how it looks once it is filtered to Type Item.

Graphical user interface, application

Description automatically generated

1. SalesLine.SETRANGE(SalesLine."Document Date", CALCDATE('<-1W>', NewDate), NewDate); The final Filter makes sure that only Orders over the period of one week are counted. “NewDate” is a variable that is set at the start of the code unit and grabs todays date which is taken from the work date on the system. I.E, “NewDate := WORKDATE”. The CALCDATE calculates the date between 2 periods Which is NewDate(Todays Date) – 1 week and NewDate.



This Process is carried out twice. Once on the SalesLine Table and once on the SalesLineArchive Table. The reason to check both is to see if any were removed. This can happen if the order is invoiced or even cancelled. There can be archives of still live orders if changes were made so the results from the SalesLine table are used to make sure that we are not pulling the same record twice. I.E – “IF NOT T\_SalesLine.FIND('-') THEN BEGIN” So this will only pull the record if it doesn’t already exist in our temporary table.

Once all the records relevant are pulled they are exported to an excel spreadsheet. Image below

Table

Description automatically generated

As you can see from the image above, we did not export all the columns as there was no need to do so. The columns we need are there. Order number, Qty and Unit Price.

From this we can use three statistical methods of validating the Data. But first we need to extract the data from the report.

There is another page on the report called Finalised Orders. This can also be drilled through to from the Pie charts on the main dash and will filter based on which one you drilled through from.

The page itself looks like this.

Graphical user interface, application, table, Excel

Description automatically generated

The data on this report is pulled from Azure which uses a relational SQL database. The main bulk of this report is from the “OrdersTakenLine” table. To see the raw data in this table we need to run a simple SQL Query.

“SELECT \*

FROM [rpt].[OrdersTakenLine\_test2]”

The above will then retrieve the below.

Table, Excel

Description automatically generated

The Finalised Order data can be extracted straight into excel by clicking “Export” and then “Analyse in excel”. Now we have reached this point we can talk about the 3 ways we can validate the data. The first is simply a count of the unique Sales order Numbers in each list. The second is a SUM of the total value of orders taken, the third is to SUM the Quantity of units sold on both extracts and see if they match.

This brings us to the next section which is dealing with the issues involved in validating this data.

1. Validation Issues and Resolutions

The biggest issue we faced when comparing the data is that the source system extract is generally a more up to date list as the Report itself only refreshes every 60 mins.

To try and get around this, we stopped the ETL process and then ran both the extract from the source system and the ETL process at the same time. This did not eliminate the problem and the Source Systems extract would still show more orders than the report.

So, we then did a vlookup so that we could find the orders that were not showing on the report.

After we reran the ETL Process to see if those orders would appear. Once the process was finished, we then took the list of missing Sales Orders and filtered for them on the report. Once they were showing and the values matched the data could be validated.

Another issue was that I am not fully fluent in the C/AL Language which is used in the Source system. That and the fact that the system itself is from 2009, it was hard to find resource materials relating to the code. Yes, there is a lot of C/AL References online, but new functions have been added since which do not work on the older systems.

Luckily The Head of IT at the time could write code in C/AL so I was able to tap into his knowledge whenever I got stuck.

1. Report Analysis

The report is predominantly used by the Sales Directors to keep a track of orders coming in that day with the ability to look back a week and see which days were the best performance that week and compare that to what offers we had available at the time to see if they made a difference to sales.

Another use of the report is a diagnostic analysis to see how many orders were cancelled or even part cancelled. From this they can see which cancelation reasons were the most common reason for a cancelation.

Chart, pie chart

Description automatically generated

Pie chart

Description automatically generated with medium confidence

You can see clearly from the above 2 screen shots that the most common reason code given on this day was BACKORDER. BACKORDER means the customer does not accept back orders, so if we cannot fulfill the full amount the customer will cancel the order and may raise a new order with the correct amount, or the order will just be part cancelled.

We can then use a filter on the Finalised orders page for BACKORDER or any other reason code to see which orders had been cancelled. See image below.

Graphical user interface, application, table, Excel

Description automatically generated

We can also drill into other levels on this page to see by Customer, Sales Person, Brand, IPG and Item.

If you hover over the sections on the Pie Chart it will give you tool tips on how much the cancelations were worth and the qty of stock. As shown below.

Text

Description automatically generated with low confidence

One way to maybe reduce the number of backorder cancelations is to provide a more regular stock report to the customers so that they do not over order. Or provide them with one if they don’t already receive one.

This however will not remove the issue entirely as stock gets picked and shipped all day, but it can help to reduce the amount of these types of cancelations saving both companies time and effort.

1. Conclusion

We have now discussed the report itself and the statistical data that it provides. Giving the sales directors a view of how many orders we have taken over the course of a week. The data in the report is from direct queries so the report is updating whenever you refresh your browser.

This gives the sales directors an almost live representation of what is coming in. The sales Team are also able to drill through to the Finalised orders page so that they can see the actual order lines.

The report also gives them a diagnostic way of measuring if a planned marketing campaign was successful by looking back at that day’s sales, or week.

We then discussed how the data was validated using a custom code unit within the source system. Then comparing the results to the report, itself.

As mentioned earlier there were issues validating the data. The biggest of these was the timing issue between the report and the code unit. The resolution of which was to refresh the report again and then see if the missing sales orders had come through.

Then we went on to talk about the analysis of the report and the ways in which it was used. These were to see the effectiveness of sales daily, Measure the success of marketing campaigns and to try and manage the cancelations.

The report itself is fully automated. Using ETL processes and Direct queries to refresh the data. Allowing the sales directors to have an almost real time view of orders coming through the system.