The end rule not populating is no longer an issue and we have analysed and better understand the way Billing is selecting data from month to month re REEFER time based charges.

If we invoice and then finalise then much of the issues with charges going to the correct month work well as configured.

We CANT finalise these reefer monitoring invoices due to the groovy part of the issue.

See below...

The groovy that CALLAO wrote to get the PUBLIC HOLIDAYS from the Calendar in Billing continues to search and apply invoice items for container having reefer connect charges AFTER

The container left.

The groovy has only one TO Date and that is the Paid through Date that the Billing user enters when running the Reefer Monitoring invoice.

So if we invoice some reefers.

Say 5

They all arrived on 01 April

We invoice on 20th April paid thru date the same

- > 3 left on 18th April which is Easter Monday in the calendar
- > 2 left on 15th April Which is Easter in the calendar
- The first 3 in the invoice correctly get 2 holidays applied to the invoice item calculations
- However the last 2 which were only here for ONE holiday also get charged both holidays.
- The same issue applies to the SATURDAY and SUNDAY counts

This means the groovy continues to count until PTDate and does not see the container TimeOut.

We need a groovy programmer to analyze the following 3 code extensions:

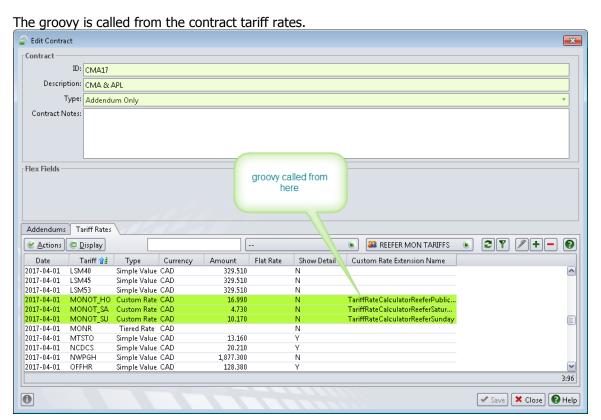
- 1. TariffRateCalculatorReeferPublicHolidays
- 2. TariffRateCalculatorReeferSaturday
- 3. TariffRateCalculatorReeferSunday

in Navis Billing

And show us how to amend it to consider the unit facility visit Time out when going to the calendar to see which billable days fall on a public Holiday.

Based on the above the groovy programmer (if they know billing code) should be able to understand when going through the groovy where it is getting the to dates.

My understanding is it is only from the paid through date entered in the invoice at generation time.



The effect is visible in the reefer monitoring invoices as follows.

