

EFT Errors – Lock Errors and Orphans

09/13/2024 6:07 pm EDT

Internal -

There is currently an issue in the eft processing service that is erroneously locking payment batches and not releasing the lock. Development is still looking into the source of the issue. Until it is fixed we expect occurrences of this issue to increase. The lock has the following consequences –

1. EFT Process locks the AR payment batch that is being used for today's approved eft transactions, and the lock gets stuck.
2. User creates a new eft transaction and selects Submit Now.
3. SedonaOffice sends the transaction to forte and it comes back with the status of Approved.
4. Forte proceeds to process the transaction as normal.
5. Back in SedonaOffice, we try mark the transaction as approved and create the deposit check, but in order to do this we have to add the check to the locked payment batch. This can't be done, the process therefore fails, and the transaction rolls back.
6. Since SedonaOffice couldn't create the check, it sends the eft transaction to Orphaned.
7. If the customer is on 6.2.16 or earlier that Orphaned transaction is now stuck and must be updated through the backend. On versions 6.2.17 and higher SedonaOffice will attempt to update the transaction and re-add the check the next time the eft service runs (which happens once an hour).
8. If the customer is on version 6.2.17 and the system attempts to correct the orphan, it will continue to fail until the lock is released. And regardless of what version the customer is on, new transactions will continue to go to orphaned until the lock is released.

When this happens support must release the lock through the data. This script will help you find the lock. Note that there may be multiple results for this query. The lock causing the issue will be dated for today, and the batch description will NOT have FND in it. FND indicates a funding batch. A lock on a funding batch will not cause new transactions to orphan, only a lock on the current NON-funding batch will cause this issue.

```
select * from SS_LockTable where lockedbyuser = 'sedonapaymentprocessing' or lockedbyuser = 'sedonaeftservice'
order by locktable_id desc
```

Once the offending lock has been identified its ID can be added to the below query and it can be deleted –

```
//delete SS_LockTable where locktable_id =
```
