

# Rectifying Lockbox Errors Programmatically – A Customer’s Practical Example

Author: Anil Patil

Created: May 9, 2007

Product: Oracle Accounts Receivable

Version: 11.5.10

## Overview

The purpose of this document is to explain how we rectified some of the recurring Lockbox Errors our client encountered using a custom concurrent program.

The users run the Lockbox process daily. Many receipts are rejected because either the lockbox data is incorrect or it conflicts with the Customer Setup data. The users have to manually correct the lockbox transmission data, which is a time consuming activity. There was therefore a need to write a small magic program (magic – because that’s what my client calls it) that would do the data rectification and clear most of the Lockbox errors. We created a custom concurrent program that the users ran after the first run of the Lockbox process.

The details of the different types of recurring errors encountered and our solution for the same is as follows

### Issue 1: MICR Number refers to more than one customer

We get this error when more than one customer record in our database have the same MICR Number that is passed in the Lockbox file. To rectify the errors manually, the users would query for the transmission data in the ‘Maintain Transmission Data’ form ( Navigation: Receipts > Lockbox > Maintain Transmission Data) and clear the MICR number ( Transit Routing and Customer Bank Account fields) for each Payment record having this error.

### Solution:

The solution we implemented for this was programmatically replicating exactly the same manual steps that the users performed.

1. We removed the MICR number details from the ‘Payment’ record from the ar\_payments\_interface table.
2. In addition to this we also gave an output report to the users with
  - a. The list of customers having the same MICR Number.
  - b. The list of receipts created after rectifying the issue.

The MICR number consists of two segments.

The first segment is the transit routing number of the customer's bank account. The transit routing number can be seen in the Banks form ( Navigation: Setup > Receipts > Banks. Field : Bank Branch Number).

The second segment is the Bank Account Number of the customer. This bank account number is associated with the customer record (Navigation: Customers > Standard. Tab: Bank Accounts. Field: Account Number).

The above issue occurs because

1. There are more than one customer defined in the system that has the same MICR number i.e. the same bank branch number and the same account number.
2. The Lockbox file only contains the Payment record with the MICR number. If the Lockbox file contained an 'Overflow Payment' record with a valid transaction number and customer number, the receipt would have been created without any lockbox errors, even if we had more than one Customer with the same MICR number.
3. There is no customer number in the 'Payment' record of the Lockbox file. If the customer number is included in the 'Payment' record of the Lockbox file, Lockbox process will ignore the MICR number and create the receipt for that customer.

Identifying the correct customer for that receipt requires some investigation on the part of the user and this takes time, hence the best practice we followed is to remove the MICR number from the lockbox file and process the Lockbox. This will create an Unidentified Receipt because we do not have any customer number or MICR number reference to link the receipt to a customer. The user then investigates for the correct customer for this receipt and updates the receipt with the customer information. Since the concurrent program output also gives the list of customers having the same MICR, the users then update the Customer record and end date the MICR number for the incorrect Customer

Scripts used in the program

1. Script to identify the customers having the same MICR number for records in the interface table

```
select bau.customer_id, cust.account_number,
bau.bank_account_uses_id, bau.start_date, bau.end_date
from ap_bank_account_uses bau,
ap_bank_accounts ba,
ap_bank_branches bb,
hz_cust_accounts cust,
ar_payments_interface int
where
    bau.customer_id = cust.cust_account_id
and bb.bank_num = int.transit_routing_number
and bb.bank_branch_id = ba.bank_branch_id
and ba.bank_account_num = int.account
and ba.bank_account_id = bau.external_bank_account_id
and int.transmission_request_id = &request_id
and int.status = 'AR_PLB_BAD_MICR_NUM'
```

```

and int.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
  AND F.record_type_lookup_code = 'PAYMENT'
  AND T.transmission_id = int.transmission_id
);

```

## 2. Script to remove the MICR number from the interface table

```

update ar_payments_interface I
SET I.transit_routing_number = NULL , I.account = NULL
where I.transmission_request_id = &request_id
and I.status = 'AR_PLB_BAD_MICR_NUM'
and I.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
  AND F.record_type_lookup_code = 'PAYMENT'
  AND T.transmission_id = I.transmission_id
);

```

Note: Records in the interface table with this issue have the status 'AR\_PLB\_BAD\_MICR\_NUM'

## 3. Script to identify customers with the same MICR Number by providing the MICR number as the parameter

```

select bb.bank_num , ba.bank_account_num, bau.customer_id,
cust.account_number, bau.bank_account_uses_id, bau.start_date,
bau.end_date
from ap_bank_account_uses bau,
ap_bank_accounts ba,
ap_bank_branches bb,
hz_cust_accounts cust
where
  bau.customer_id = cust.cust_account_id
and bb.bank_num = '&transit_routing_number'
and bb.bank_branch_id = ba.bank_branch_id
and ba.bank_account_num = '&bank_account_number'
and ba.bank_account_id = bau.external_bank_account_id ;

```

## Issue 2: MICR and Customer Number do not refer to the same customer.

We get this error when

1. The lockbox file contains the 'Payment' record as well as the 'Overflow Payment' record
2. The MICR number of the 'Payment' record points to one customer and the customer number of the 'Overflow Payment' record is a different customer.

To rectify the errors manually, the users would query for the transmission data in the 'Maintain Transmission Data' form (Navigation: Receipts > Lockbox > Maintain Transmission Data) and clear the MICR number (Transit Routing and Customer Bank Account fields) for each Payment record having this error.

## Solution:

The solution that we implemented for this was

1. Programmatically remove the MICR number for the 'Payment' record from the interface table. Once we remove the MICR number (transit\_routing\_number and the account number), there is no field that links the 'Payment' record to the Customer record and hence the receipt gets created for the customer that appears on the 'Overflow' Payment record.
2. Provide the users with a report that gives them the MICR number and the corresponding Customer number, the receipt number and the customer of that receipt. This report helped the users to verify the MICR number of the Customer.

Script to remove the MICR Number

```
update ar_payments_interface I
SET I.transit_routing_number = NULL , I.account = NULL
where I.transmission_request_id = &request_id
and I.status = 'AR_PLB_CUSTOMER_CONFLICT'
and I.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
  AND F.record_type_lookup_code = 'PAYMENT'
  AND T.transmission_id = I.transmission_id
);
```

Note: Records in the interface table with this issue have the status 'AR\_PLB\_CUSTOMER\_CONFLICT'

## Issue 3: Receipt has more than one customer number

We get this error when

1. There are more than one 'Overflow Payment' records in the lockbox file.
2. All the 'Overflow Payment' records do not have the same Customer number.

Rectifying these errors manually was a very time consuming activity as the users had to remove the MICR number and the customer number from the 'Payment' and the 'Overflow Payment' records. As a result of this, the receipts would always be created as unidentified receipts.

## Solution:

The Solution we implemented for this was

1. If the 'Payment' record has a valid MICR number, then update the customer numbers from the 'Overflow Payment' records with the Customer number having the MICR Number of the 'Payment Record'. Most of the records having this issue satisfied the condition of 'Valid MICR Number' for the client.

Script to get the account\_number

```
select cust.account_number
into v_account_number
from ap_bank_account_uses bau,
ap_bank_accounts ba,
ap_bank_branches bb,
hz_cust_accounts cust,
ar_payments_interface int
where
    bau.customer_id = cust.cust_account_id
and bb.bank_num = int.transit_routing_number
and bb.bank_branch_id = ba.bank_branch_id
and ba.bank_account_num = int.account
and ba.bank_account_id = bau.external_bank_account_id
and int.transmission_request_id = '&request_id'
and int.status = 'AR_PLB_CUST_NUM_CONFLICT'
and int.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
    AND F.record_type_lookup_code = 'PAYMENT'
    AND T.transmission_id = int.transmission_id
);
```

Note: When the receipt has more than one customer number, the status in the interface table is 'AR\_PLB\_CUST\_NUM\_CONFLICT'

Script to update the 'Overflow Payment' records with the same customer number derived from the above query

```
update ar_payments_interface I
SET I.customer_number = v_account_number
where I.transmission_request_id = &request_id
and I.status = 'AR_PLB_CUST_NUM_CONFLICT'
and I.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
    AND F.record_type_lookup_code = 'OVRFLW PAYMENT'
    AND T.transmission_id = I.transmission_id
);
```

2. If the 'Payment' record does not have a valid MICR Number ( i.e. No customer have that MICR Number) or if the MICR Number is blank, we cleared the

customer number for all the 'Overflow Payment' records. This approach would create an unidentified receipt which the users would later investigate.

Script to clear the customer number's of the 'Overflow Payment' records

```
update ar_payments_interface I
SET I.customer_number = NULL
where I.transmission_request_id = &request_id
and I.status = 'AR_PLB_CUST_NUM_CONFLICT'
and I.record_type =
( select record_identifier
  from ar_trans_record_formats F , ar_transmissions T
  where F.transmission_format_id = T.requested_trans_format_id
  AND F.record_type_lookup_code = 'OVRFLW PAYMENT'
  AND T.transmission_id = I.transmission_id
);
```

3. We provided the users with the list of receipts that were rectified using this approach along with the corresponding invoice numbers and the original customer numbers as they appeared before the interface tables was updated.

## Summary:

This document details the approach we took for rectifying the recurring Lockbox errors and this saved 2 to 3 hrs per day of the Cash Apps Analyst.

The type of Lockbox errors different customers get while processing the Lockbox would be different based on how they receive the Lockbox file from the bank. My approach would always be to discuss the most recurring issues with the Cash Apps Team members and provide them a custom program that would do the rectification.

There is no better alternative than to get the proper lockbox file from the bank that would never error. However, in reality this seldom happens and in those cases, this approach WORKS !!!

## References:

Note 282749.1 ARLPLB Lockbox Status Error Codes

Note:199014.1 How to identify Receipts on basis of Bank Account Number in Lockbox Process?

Note:301133.1 Autolockbox Fails Validation With MICR Number Refers To More Than One Customer

## Appendix:

The transmission formats used in the examples above are

### Transmission format for 'Payment' Record

| Position |     | Field Type           | Justify | Fill Symbol | Date | Time | Overflow Indicator |        |
|----------|-----|----------------------|---------|-------------|------|------|--------------------|--------|
| Start    | End |                      |         |             |      |      | Format             | Amount |
| 1        | 1   | Record Identifier    | Right   | Blank       |      |      |                    |        |
| 2        | 4   | Batch Name           | Right   | Zero        |      |      |                    |        |
| 5        | 7   | Item Number          | Right   | Zero        |      |      |                    |        |
| 8        | 17  | Remittance Amount    | Right   | Zero        |      |      | Yes                |        |
| 18       | 26  | Transit Routing Numb | Left    | Blank       |      |      |                    |        |
| 27       | 40  | Account              | Left    | Blank       |      |      |                    |        |
| 41       | 50  | Payment Number       | Left    | Blank       |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |
|          |     |                      |         |             |      |      |                    |        |

### Transmission Format for 'Overflow Payment' Record

| Position |     | Field Type         | Justify | Fill Symbol | Date | Time | Overflow Indicator |        |
|----------|-----|--------------------|---------|-------------|------|------|--------------------|--------|
| Start    | End |                    |         |             |      |      | Format             | Amount |
| 1        | 1   | Record Identifier  | Right   | Blank       |      |      |                    |        |
| 2        | 4   | Batch Name         | Right   | Zero        |      |      |                    |        |
| 5        | 7   | Item Number        | Right   | Zero        |      |      |                    |        |
| 8        | 10  | Overflow Sequence  | Right   | Zero        |      |      |                    |        |
| 11       | 11  | Overflow Indicator | Right   | Zero        |      |      | 0                  |        |
| 12       | 26  | Invoice 1          | Left    | Blank       |      |      |                    |        |
| 27       | 29  | Attribute 4        | Left    | Blank       |      |      |                    |        |
| 30       | 37  | Customer Number    | Right   | Zero        |      |      |                    |        |
| 38       | 47  | Amount Applied 1   | Right   | Blank       |      |      | Yes                |        |
| 49       | 50  | Attribute 5        | Left    | Blank       |      |      |                    |        |
|          |     |                    |         |             |      |      |                    |        |
|          |     |                    |         |             |      |      |                    |        |
|          |     |                    |         |             |      |      |                    |        |