WITN04640102

EUM Locking Issues Technical Investigation

v0.18 21/12/2018

Author: Jon Hulme

1. Summary of the problem

The End User Management (EUM) project was released into pilot at the end of July 2018, and the functionality was subsequently rolled out across the estate.

This changed a fundamental assumption of the Horizon design, which is that a single Horizon user can only be logged in once at any one point in time. If the Horizon user attempts to login concurrently on a second counter, then a "concurrent login" message is output and Horizon will forcibly terminate the previous user session.

The change was to allow a given EUM user to lock a counter and then log on to another counter. The user can later lock or log off from this counter and eventually resume the user session on the original counter. There can be multiple sessions in a locked state, but only one user session actually active at any point in time.

This has caused incidents to occur in the live estate (e.g. PC0275532, PC0275563) if the active user session rolls over the current stock unit, and a locked session on another counter attached to the same stock unit is then resumed. This is because when the locked user session is resumed the counter is not aware of the rollover and continues to trade in the old trading period (TP) / balance period (BP). These transactions are then recorded in the old TP/BP and so will not appear on the branch accounts, although they will be successfully sent to back end systems, and will be visible in the counter transaction log for the old TP/BP.

Further, if the branch is rolled over on the resumed counter, then a non-zero trading position will result and carried forward suspense figures may be incorrect.

Investigation has shown that they are other scenarios that may be misleading (see below), but none of these have been found to lead to balancing errors. It is possible that there could be other scenarios that might cause problems. If so, they will be resolved using the normal processes for incident reporting and handling.

Solution options are discussed in section 3.

WITN04640102

2. Technical Description of the Issue

The ability for an EUM user, i.e. one with a Horizon user (HUID) that is linked to a Post Office User Id (POID), to lock a counter and then be able to log on to another counter has led to the following issues.

Further investigation is required to determine whether there are other issues.

1. Unlocking a counter can lead to it trading in the wrong TP/BP.

   Consider the scenario:
   a. An EUM user locks counter A.
   b. The same user logs into counter B.
   c. Counter B rolls over the stock unit.
   d. Counter B is locked or logged off.
   e. The same user unlocks counter A.

   Now counter A is not aware that the stock unit has rolled over, and will now trade in the old TP/BP, until it logs out.

   This scenario has occurred in the live estate. The SSC can detect when it has happened by looking for transactions that have occurred for a given TP/BP after the date/time on which that TP/BP rolled over.

   A variant of this scenario is that counter A times out due to inactivity, rather than being unlocked, after counter B has rolled over. Since timeout auto settles any items in the basket to cash, in this scenario any items in the basket would be settled in the old TP/BP.

   Another variant of this scenario is that if the clerk proceeds to rollover the branch, then a non-zero trading position is reported and the trading statement produced is incorrect (notably the Carried Forward cash position is identical to the Brought Forward cash position). The only financial figures written by the branch rollover process are the branch suspense opening balances for the trading period rolled into, which may be incorrect if a non-zero net value is present in suspense.

   The System Support Centre (SSC) have run a check to determine branches impacted by the above scenarios since EUM went live, and found a number of cases, the details of which have been passed to Post Office Limited. See the associated SSC report.

2. Logging into to a second counter removes the balancing lock from a stock unit.

   Consider the scenario:
   a. An EUM user logs in to counter A.
   b. The user presses the button to balance the stock unit.
   c. The stock unit is locked.
   d. The balancing process is started.
   e. The user locks the counter.
   f. The same EUM HUID user logs in to counter B.

WITN04640102

g. This unlocks the stock unit since the user is the same as the locking user.
h. The user may trade in the stock unit since it is not locked for balancing, and could even rollover the stock unit.
i. The user locks or logs out from Counter B.
j. The user unlocks counter A.
k. The user resumes stock unit balancing.

Investigation shows that this causes a system error message and fails the stock unit rollover because the system detects the stock unit is not locked.

Note however that the system error is not output until the end of the rollover process, and so:

i) A balance report is printed prior to the rollover failure.
ii) The financial position reported by the rollover process, including declarations, discrepancies and making good local suspense may not be based on the latest position in the stock unit.

Note that if a user with a different HUID (even if linked to the same POID) attempts to login while the first session in the above scenario is locked, they are correctly unable to login since the rollover lock present.

The SSC have run a check to see if there were any duplicated stock unit rollovers for same TP and BP since EUM went live, and none were found (except for those that has rolled over so many times in a short period that the TP was re-used).

3. Logging into to a second counter removes the balancing lock from the branch.

Consider the scenario:

a. An EUM user logs in to counter A.
b. The user presses the Trading Statement button to balance the branch.
c. The branch is locked.
d. The branch balancing process is started.
e. The user locks the counter.
f. The same EUM HUID user logs in to counter B.
g. This unlocks the branch since the user is the same as the locking user.
h. The user rolls over the branch into a new trading period.
i. The user locks or logs out from Counter B.
j. The user unlocks counter A.
k. The user resumes branch balancing.

Investigation shows that this fails because the system detects the branch is not locked.

Note that the failure does not occur until the end of the branch rollover process, and:

i) The counter does not report failure to the clerk, but rather reports successful rollover into the next trading period, even though the rollover actually failed. This has been raised as defect PC0275644.
ii) An invalid branch trading statement is printed prior to the rollover failure.

WITN04640102

Note that if a user with a different HUID (even if linked to the same POID) logs in while the first session in the above scenario is locked, they are correctly unable to rollover the branch since the branch rollover lock is present.

The SSC have run a check to see if there were any duplicated branch rollovers for same TP since EUM went live, and none were found (except for those that has rolled over so many times in a short period that the TP was re-used).

WITN04640102

3. Proposed solutions

Note that a user session can be in one of the following states:

- NEW: The session is attempting to login (temporary state).
- ACTIVE: The session is active as the user is logged in.
- LOGOUT: The session has been ended as the user logged out.
- INVALIDATE: The session has been ended as the user failed to login.
- FAILED: The session was abnormally ended, e.g. when terminated due to concurrent login.
- RECOVERING: The session is being recovered by counter recovery.

3.1 Option 1: Remove Lock Functionality

We could change the lock button on the counter to no longer apply the lock centrally. This means that if the same HUID, or any HUID linked to the same POID, attempts to log on elsewhere then they will only be able to logon if they opt to terminate the previous user session (this is the normal "concurrent logon" functionality).

This would prevent further issues occurring, and so contains the problem to the point at which this change is made. It is the simplest change to make, with the least risk of side effects. It is a data centre only change that could be issued as a hot fix.

It does however mean that the same Drop & Go user would not be able to lock a Drop & Go session and then use a different counter to serve a customer.

Post Office Limited have to date not wanted to pursue this option.

3.2 Option 2: Prevent the same HUID logging in concurrently

We could disallow the same HUID from logging in to another counter. A different HUID linked to the same POID could still login to another counter.

This means that if the same HUID attempts to log on elsewhere then they will only be able to logon if they opt to terminate the previous user session (this is the normal "concurrent logon" functionality).

This would prevent further issues occurring, and so contains the problem to the point at which this change is made. It is more complex that Option 1 and so would require greater testing. It is a data centre only change that could be issued as a hot fix.

It does however mean that a Drop & Go user would need two HUIDs linked to the same POID to be able to lock a Drop & Go session and then use a different counter to serve a customer.

Post Office Limited have to date not wanted to pursue this option.

WITN04640102

3.3 Option 3 – Add restrictions to back office functionality when lo

This option addresses the locking issues related to the same HUID linked to a POID logging in concurrently.

The following sub-sections list the scenarios that have been identified to be potentially problematic, and the recommended solution.

3.3.1 Stock Unit Balancing

3.3.1.1 Unlocking a counter after the same HUID logged in on another counter and rolled over the stock unit

This scenario has occurred in the live estate.

The fix is to prevent stock unit balancing being started if the same HUID is logged on elsewhere, even if that other session is locked.

The code already prevents stock unit balancing being started if a different HUID is logged on elsewhere, even if that other session is locked.

When the "Balance Report" button is pressed, the fix is to check for other user sessions associated with the stock unit that are in the ACTIVE, FAILED, or RECOVERING state, including the current user's sessions, irrespective of whether the sessions are locked. If any are found then display the existing message MSG31305 "Unrecovered Sessions" and abort the stock unit balancing attempt.

This does mean that if a counter was balancing and physically failed, it would need to be replaced before balancing could continue. That is however the current position since failed user sessions associated with a stock unit must be recovered before stock unit balancing.

This fix requires a counter, data centre and reference change, and will need SV&I testing.

When a counter is unlocked, a further check should be performed to ensure that the current TP, BP and attached stock unit are those that the counter has cached in memory (see also section 3.3.3). If they are different then a system error should be generated and counter force logged out.

This check also needs to be applied if the system times out at the lock screen and attempts to auto-settle an item that has been left in the basket.

3.3.1.2 Logging in at a counter while the same user HUID on another counter is balancing the stock unit

The fix in overview is to prevent a user logging in if that same user is balancing the stock unit on another counter, unless they terminate the balancing session.

WITN04640102

The code already prevents a different HUID (whether the same POID or not) from logging in while the stock unit is being balanced.

In more detail, the fix is to add a new check for the case where a user attempts to login, and the stock unit is locked by that HUID linked to a POID, and there is an ACTIVE session present for that HUID on another node. In this case display a new message warning that the stock unit is locked for balancing, and to continue and unlock the stock unit then all other sessions attached to this stock unit will be terminated (there should normally only be the one session for the same HUID on another node that started the balance).

This fix requires a counter, data centre and reference change, and will need SV&I testing.

Allowing the user to terminate other sessions and login means that it is possible for the user to continue trading if a counter that the user is balancing physically fails.

3.3.1.3 Unlocking a counter (i.e. with the same HUID) while another counter is balancing the stock unit and is locked

This scenario will be prevented by the fix to 3.3.1.1 and 3.3.1.2 above, because a user cannot start to balance the stock unit if there are other active user sessions for that stock unit, and a user cannot login if another session is balancing the stock unit.

3.3.2 Branch Balancing

3.3.2.1 Unlocking a counter after the same HUID logged in on another counter and rolled over the branch

The counter does not cache the branch trading period because it is always possible for another user to rollover the branch. So this is not an issue.

3.3.2.2 Logging in at a counter while the same user HUID on another counter is balancing the branch, and then proceeding to balance the branch

This is a scenario that allows two counters to attempt to balance the branch simultaneously.

The code already prevents a different HUID from balancing the branch if it is already being balanced on another node.

Strictly, we could allow a user to login to another counter even if that same HUID user was balancing the branch and locked on another node, and just disallow that user from balancing the branch. However, the problem is that that user's second login removes the balancing lock and does not have the option of cancelling the original balancing session in the case of that counter being unusable.

The best solution is the same as that for the stock unit balancing lock, i.e. to prevent a user logging in if that same user HUID is balancing the branch at another node, unless they terminate the balancing session.

WITN04640102

In more detail, the fix is to add a new check for the case where a user attempts to login, and the branch is locked for balancing by that HUID linked to a POID, and there is an ACTIVE session present for that HUID on another node. In this case display a new message warning that the branch is locked for balancing, and to continue and unlock the branch then all other sessions for this user (HUID) will be terminated.

This fix requires a counter, data centre and reference change, and will need SV&I testing.

Allowing the user to terminate their other sessions and login means that it is possible for that user continue to balance the branch if a counter that the user is balancing physically fails.

3.3.2.3 Unlocking a counter (i.e. with the same HUID) while another counter is balancing the branch and is locked

If the branch is being balanced by a different HUID, this is fine. The existing code will stop that HUID starting to balance the branch concurrently.

If the branch is being balanced on another node by the same HUID, then the fix in 3.3.2.2 above means you cannot be in this scenario as the locked session would have been terminated.

3.3.3 Attaching a user to a stock unit

When the clerk choses a user to change the stock unit they are attached to, the counter performs a check to see if that user is logged on, and gives an error message (MSG30002) if they are.

If that check passes, the clerk choses the stock unit to attach to and calls the data centre to action this. The data centre performs the following checks relevant to this document:

- Check that the user to attach is not currently balancing a stock unit
- Check that the stock unit to attach is not currently being balanced
- If the user to be attached is not the current user, check that the user is not logged on.

If the final check fails, MSG30002 is again displayed by the counter. There is a bug that in this case it does not display the user name. This has been raised as defect PC0275906.

3.3.3.1 Unlocking a counter after the same HUID on another counter changed the stock unit that the user is attached to

In this case, the unlocked counter is not aware of the change to the attached stock unit, and can continue to trade in the wrong stock unit.

The following fixes were considered:

1. To disallow a user's attached stock unit from being changed if that same HUID has an ACTIVE or RECOVERING session on another node (which would be locked).

WITN04640102

The problem with this is that if the locked counter failed then the user would not be able to unlock the counter to close down the session. Therefore the attached stock unit could not be changed until the counter was replaced and a user logged in (causing the previous user sessions to be tidied up).

2. When attempting to change the attachment, to give a warning message, and only allow the attachment to be changed if the other session is terminated (as at login). This would be the first time we terminate user sessions except at login, and introduces some risk of possible side effects.

3. When attempting to change the stock unit that a user is attached to, if other ACTIVE or RECOVERING sessions for that user are found then give a warning message, but allow the user to proceed.

   When unlocking a counter, if the attached stock unit does not match that currently known to the counter, then give an error and force logout the user.

   This is the recommended solution, as agreed with Post Office Limited. The fix requires a counter, data centre and reference change, and will need SV&I testing.

3.3.4 Modify User

3.3.4.1 Changing a User's Password

Horizon already allows a user's password to be changed, no matter whether that user is logged in or not, or whether they are the current user.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.4.2 Changing a User's Role

Horizon disallows changing a user's role if they are logged in, even if they are the current user.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.4.3 Changing a User's Disabled Status

Horizon already allows a user to be disabled, or enabled, no matter whether that user is logged in or not, or whether they are the current user, so long as they have the correct permission.

A user can still lock and unlock the counter even if their account is disabled, but they cannot login as a new session.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.4.4 Changing a User's Must Change Password Status

Horizon already allows a user to set the Must Change Password status, no matter whether that user is logged in or not, or whether they are the current user, so long as they have the correct permission.

WITN04640102

A user can still lock and unlock the counter even if the Must Change Password status is set.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.4.5 Changing a User's Unlock Account Status

Horizon allows a user to unlock a locked user account, no matter whether that locked user account is logged in or not. It not possible for the same user to unlock their account, because they could not be logged in to do so.

With EUM locking, a user cannot unlock a counter if their user account has become locked by that same HUID attempting to login on another counter and giving an incorrect password three times.

The user account would need to be unlocked by another user.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.5 Deleting a User

Horizon prevents a user being deleted if that user is logged in, no matter whether it is the same user or a different user that is attempting to perform the deletion.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.6 Deleting a stock unit

Horizon prevents a stock unit from being deleted if any user is attached to that stock unit, no matter whether it is the attached user or a different user that is attempting to perform the deletion.

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.7 Reporting and Cut-Offs

The counter already allows multiple users to view reports concurrently, and handles the case of attempting to cut off reports concurrently. The first cut off chronologically will succeed, but the second will display error message MSG00524 "Already Cut Off".

Therefore the introduction of logging in twice with the same user id should not be an issue, although this should be tested.

3.3.8 Declarations

3.3.8.1 Declarations after stock unit rollover over on second node

The declaration is recorded against the wrong TP/BP, and so will be ignored by the rollover process. Any variance/discrepancy shown will be incorrect.

WITN04640102

This will be fixed by the stock unit rollover fix described in section 3.3.1.

3.3.8.2 Declarations performed concurrently

The counter displays the following message if it finds that a declaration has already been updated at another counter, no matter whether it was the same user or not:

> MSG00081 "Concurrent Declaration" stating "Declaration <id> has already been made at another counter, do you wish to overwrite this?" With Continue or Cancel buttons.

If the Variance button is pressed, and the counter is locked at the screen which display the latest declaration date/time, and another counter then changes that declaration, when the first counter is resumed it will actually be using the latest declaration for comparison, not the one shown for the date/time shown on the screen at the time. This is the currently live behaviour which was not changed by EUM.

This is a defect which has been raised as PC0275890. Horizon should check to see if there are any new or changed declarations since it displayed their details on the screen, otherwise the variance figure is misleading. If it finds declarations have changed then I recommend a message is output explaining that one or more declarations have changed, and the declaration list screen should be redisplayed with the updated declaration details.

Post Office to confirm that this behaviour should be changed.

This is an existing issue, and is not related to EUM locking.

3.3.9 Transfers

3.3.9.1 Transfers after stock unit rollover over on second node

Transfers will suffer from the same issue as performing any transaction in the case where the counter is locked, and another counter rolls over the stock unit, then the original counter is unlocked:

- A Transfer Out on the original counter would then attempt transfer out into the wrong TP BP.

  If it was just the BP that has rolled over, then the transfer out transaction is recorded against the old TP BP (and so would be missing from the branch accounts), but can be transferred in.

  If it was the TP that has rolled over, the counter shows the list of available stock units in the old TP (which should ring alarm bells), but when the clerk tries to settle the Transfer Out then the counter gives error message MSG10104 stating that the destination TP is not the branch TP, which although not true does prevents the transfer out.

- A Transfer In would transfer into the wrong TP / BP.

Note that it a stock unit cannot be rollover into a new TP if it has pending transfers in or out, but it can rollover into a new BP.

This will be fixed by the stock unit rollover fix described in section 3.3.1.

WITN04640102

3.3.9.2 Transfers performed concurrently

Transfers do not involve user specific locking, so will not be impacted by the same user performing the transfer twice. If two counters try to perform the Transfer In concurrently, the existing transfer status checks cause the second counter to get error message MSG90991 "Transfer In Failure".

3.3.10 Remittances

3.3.10.1 Remittances after stock unit rollover over on second node

Remittances will suffer from the same issue as performing any transaction in the case where the counter is locked, and another counter rolls over the stock unit, then the original counter is unlocked, in that they will be recorded against the wrong TP/BP.

3.3.10.2 Remittances performed concurrently

If two counters attempt to record delivery of the same pouch at the same time, the second will usual give a failure at settlement time. In the rare case that the pouch details are not found in the LFS information in the data centre, then there is a defect which allows the same pouch to be delivered twice, which for Cash or ForEx pouches would cause Horizon to think the branch has more cash than in reality. This has been raised as PC0275893. It would be very unlikely to occur, since it only applies when no LFS information is present, and the one physical pouch must be scanned concurrently on two counters. This is an existing issue, and is not related to EUM locking, but it should be tested.

Remming out to the same pouch concurrently is already prevented, the second counter to settle the rem out with the same pouch id gives settlement error message MSG90955.

Using the cheques button to Rem Out and Cut Off the cheques report concurrently, does rem out the cheques twice, although the second cut off them fails with MSG00524 "Already Cut Off". The impact is that the branch will be out of pocket and needs to reverse the second rem out. This scenario is very unlikely to happen because the business process means that the clerk should physically have the cheques and dispatch them when processing, and should realise that there is one set of cheques.

This has been raised as defect PC0275902 for this scenario and recommend that the solution is to output a different message instead of MSG00524 which advises the clerk that a concurrent rem out has occurred and that the second rem out should be reversed.

Post Office to confirm that this behaviour should be changed.

3.3.11 Rolling Over Inactive Stock Units

3.3.11.1 Rollover after inactive stock unit rollover over on second node

A stock unit is not considered inactive if it has any user sessions logged in. Therefore, even if a counter is logged in and locked, the stock unit is not considered inactive, and so cannot be rolled over on a second counter.

Therefore there is no problem in this scenario, but it should be tested.

WITN04640102

3.3.11.2 Inactive Roller performed concurrently

The existing code checks that the stock unit is in the same TP and the branch TP, and if this is not true then the stock unit is not inactive and cannot be rolled over.

This prevents concurrent inactive stock unit rollover, even if the same user is concurrently performing the rollover.

Therefore there is no problem in this scenario, but it should be tested.

3.3.12 Processing Transaction Corrections

3.3.12.1 Processing TC after inactive stock unit rollover over on second node

The TC is recorded against the wrong TP/BP.

This will be fixed by the stock unit rollover fix described in section 3.3.1.

3.3.12.2 Processing TC performed concurrently

The counter already caters for concurrent processing, no matter whether it is by the same HUID or not, and the second counter to attempt the TC processing gives error message MSG01063 "Transaction Correction Unavailable".

Therefore there is no problem in this scenario, but it should be tested.

3.3.13 Processing Transaction Acknowledgements

3.3.13.1 Processing TA after inactive stock unit rollover over on second node

The TA is recorded in the correct TP/BP.

Therefore there is no problem in this scenario, but it should be tested.

3.3.13.2 Processing TA performed concurrently

The counter already caters for concurrent processing, no matter whether it is by the same HUID or not, and the second counter to attempt the TA processing gives error message MSG00346 "TA(s) Already Processed".

Therefore there is no problem in this scenario, but it should be tested.

Note that managing a stock unit association concurrently does keep the value of the last counter to make a change. This is expected behaviour.