# Configure workflow for fixing catch weight issue when picking

## Summary:

This workflow is aimed to address the issue of picking catch weight item, when user picked all cartons from the picking face location, as long as user scanned weight for these cartons does not match with system recorded, system is always giving whatever recorded regardless what user scanned, based on data setup, there are 3 different scenarios mentioned:

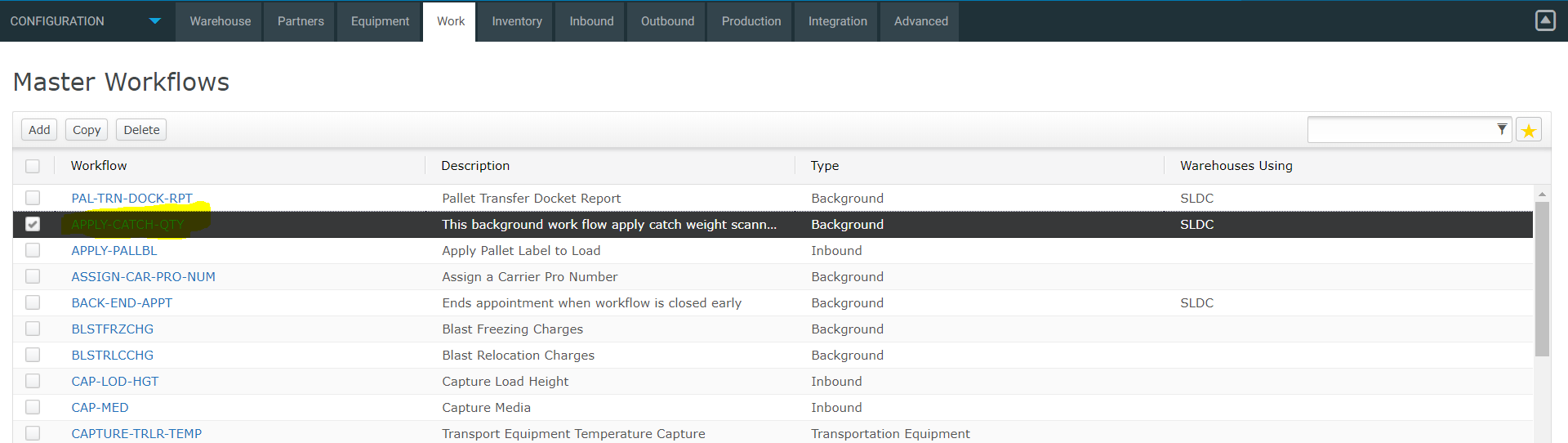
1. Pick location has 1 carton and system weight of 19.54 kg. Tolerance range 5.356 - 11.124. User confirms pick of 1 and scans weight within tolerance.
   * Result - system takes remaining weight of 19.54 kg. User is unable to continue as the system prompts "outside of tolerance".
2. Pick location has 1 carton and system weight of 11 kg (within tolerance).  User confirms pick of 1 and scans actual carton weight of 10.03 kg (within tolerance).
   * Result - system weight of 11 kg is recorded against the pick.
3. Pick location has 1 carton and system weight of 4.6 kg (within tolerance).  User confirms pick of 1 and scans actual carton weight of 5 kg (within tolerance).
   * Result - system weight of 4.6 kg is recorded against the pick.

## Approach:

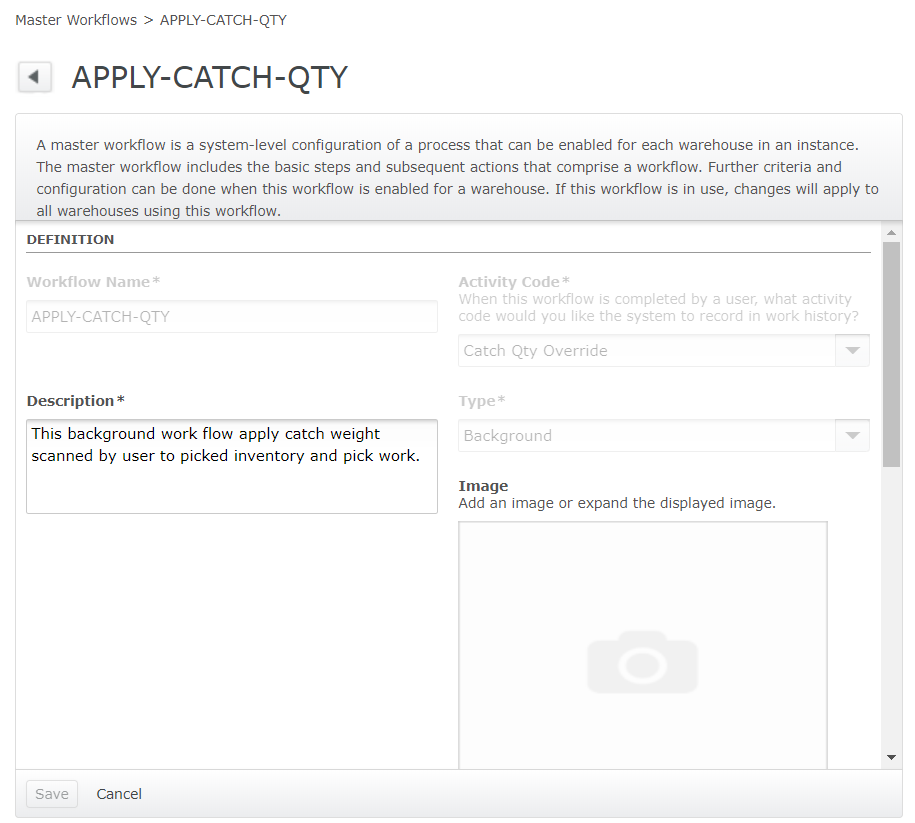
We can use ‘INVMOV’ background workflow to address these issues, basically above issues are root caused by the fact that in the backend system is always moving catch qty left on inventory to be applied without respecting catch qty scanned by user.

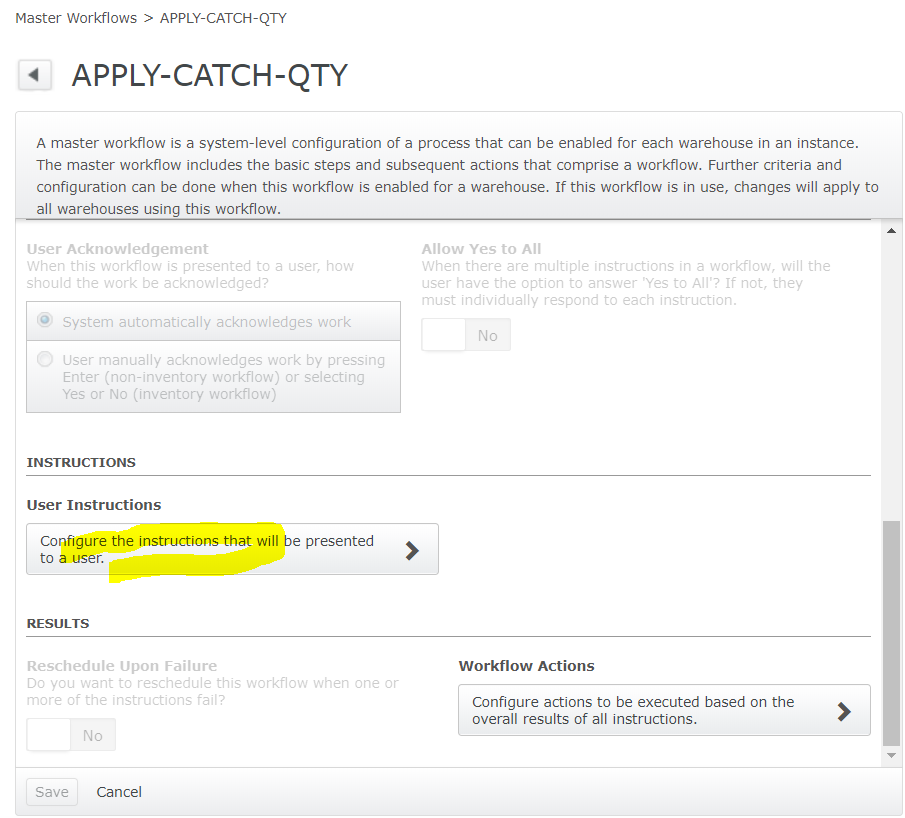
## Workflow configuration:

1. go to Master workflow to create a new workflow, named APPLY-CATCH-QTY:



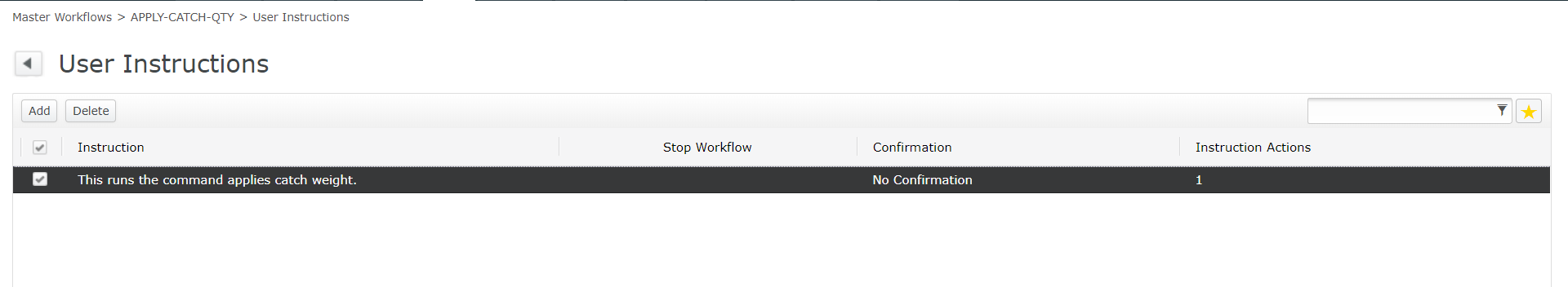
2. detail about APPLY-CATCH-QTY:



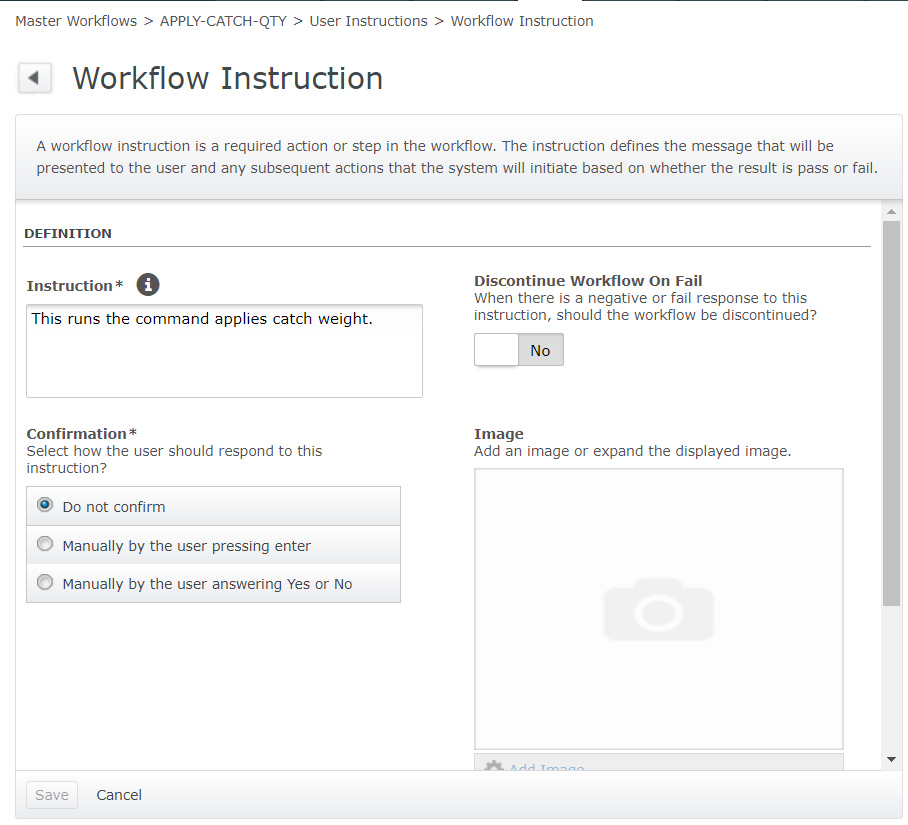


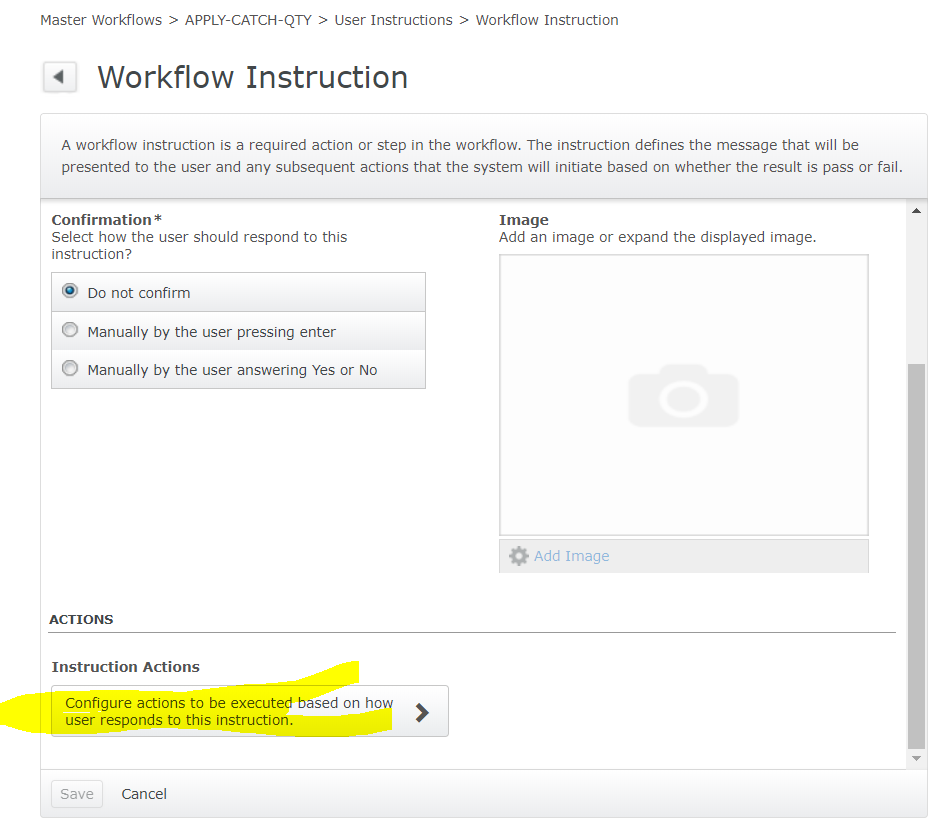
3. click ‘User Instructions’:

Add a config:



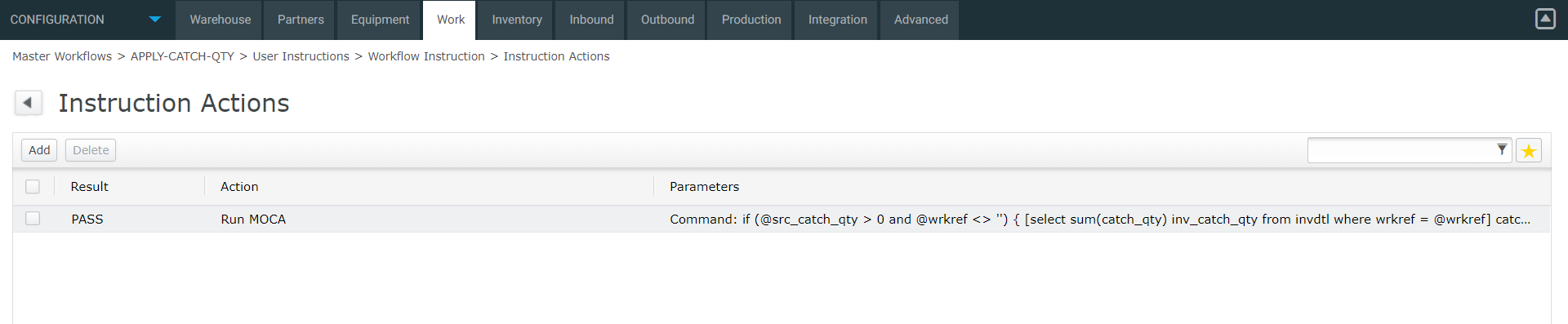
With detail:





4. Click ‘Instruction Actions’:

Add a new config:



The code to run as:

if (@src\_catch\_qty > 0 and @wrkref <> '')

{

[select sum(catch\_qty) inv\_catch\_qty

from invdtl

where wrkref = @wrkref]

|

if (@src\_catch\_qty <> @inv\_catch\_qty)

{

[select sum(untqty) totqty

from invdtl

where wrkref = @wrkref]

|

if (@totqty > 0)

{

publish data

where avg\_catch\_qty = @src\_catch\_qty / @totqty

|

[select dtlnum, untqty

from invdtl

where wrkref = @wrkref

order by dtlnum]

|

[update invdtl

set catch\_qty = @avg\_catch\_qty \* @untqty

where dtlnum = @dtlnum]

;

[update pckwrk\_hdr

set pck\_catch\_qty = @src\_catch\_qty,

app\_catch\_qty = @src\_catch\_qty

where wrkref = @wrkref]

;

[select ship\_line\_id,

sum(catch\_qty) line\_catch\_qty

from invdtl

where wrkref = @wrkref

group by ship\_line\_id]

|

[update pckwrk\_dtl

set pck\_catch\_qty = @line\_catch\_qty,

app\_catch\_qty = @line\_catch\_qty

where wrkref = @wrkref

and ship\_line\_id = @ship\_line\_id]

}

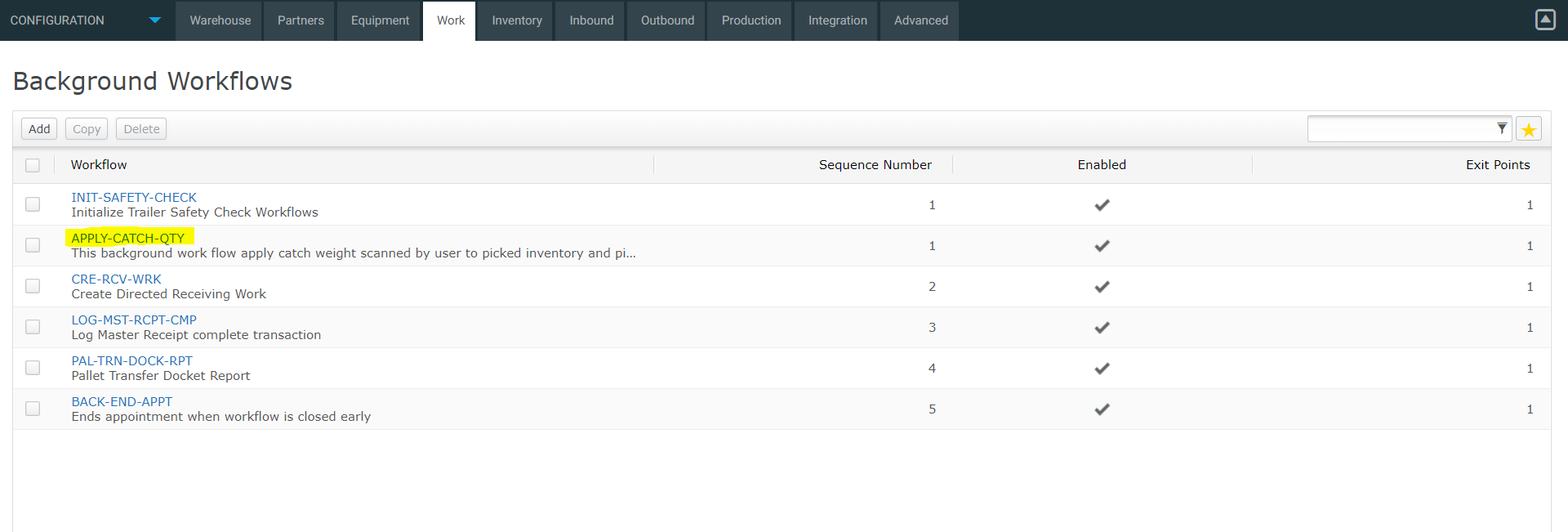
}

}

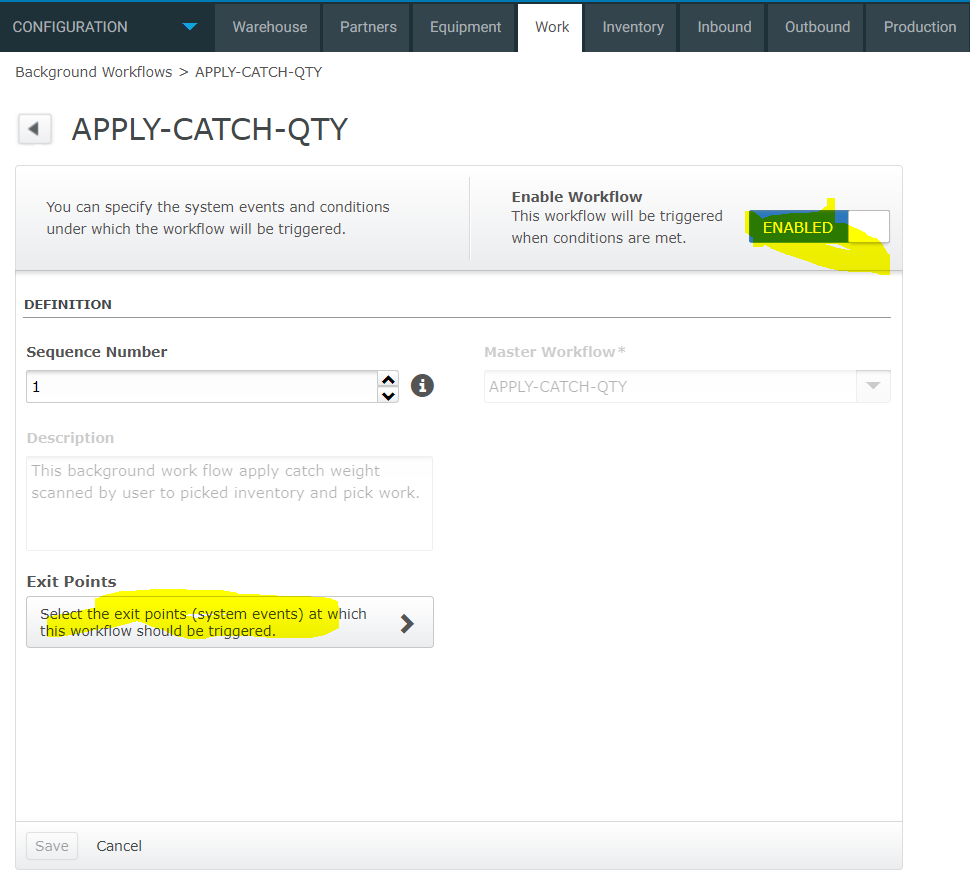
Note to above logic:

This logic will use scanned catch weight, and spread this weight to all details which are picked by the wrkref, the ‘spread’ logic is that it will get total picked quantity, for example 50 from 5 cartons(details), and user scanned weight 60KG, so weight per unit qty is 60 / 50 = 1.2KG, for each carton, it will apply weight 1.2 \* 10 = 12KG. at the end the total scanned weight will be applied to pick work header and pick work detail. Please note this code will spread weight to each carton with equal weight value, for example, all cartons will get exact same weight if they are unit quantity same. The total weight from these cartons will be equal to user scanned weight.

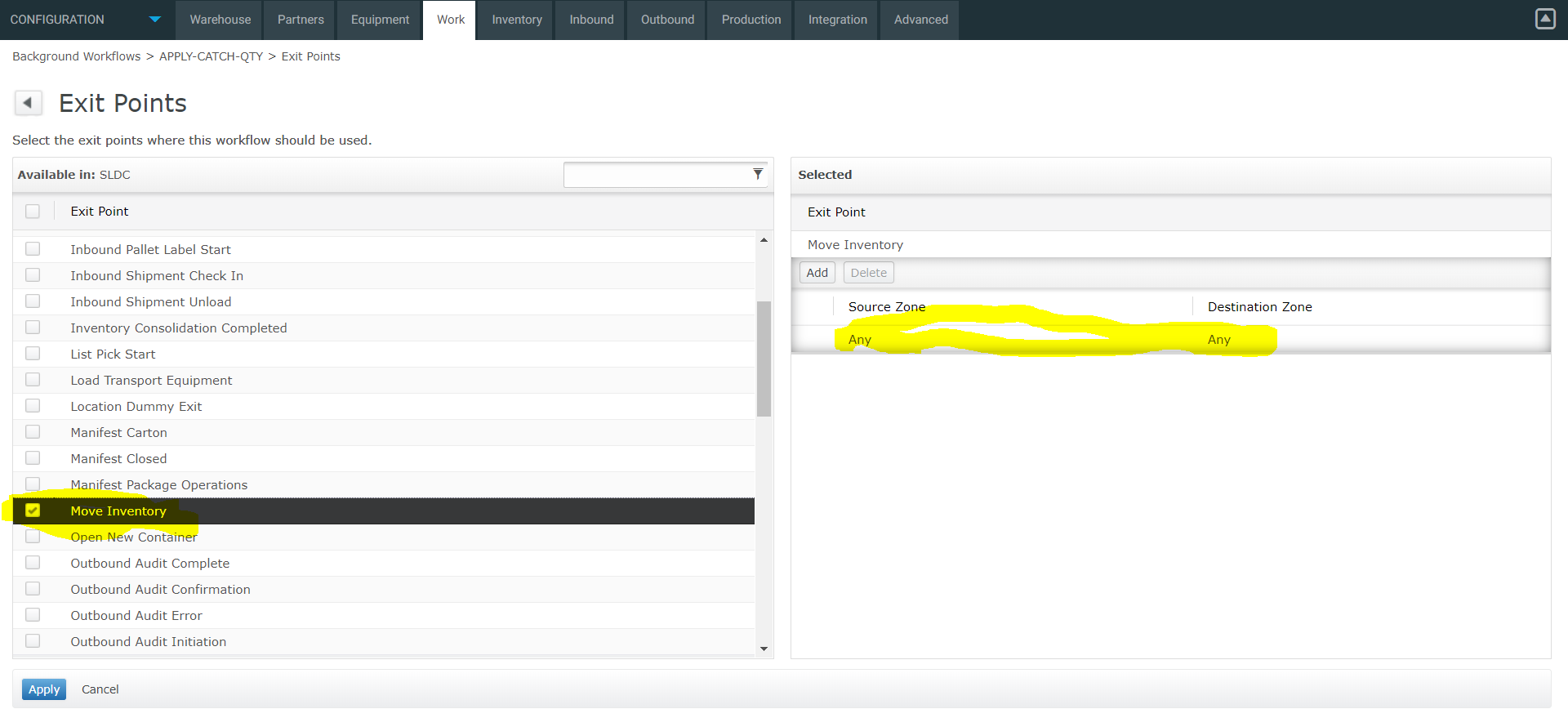
5. go to background workflow to add a new record:



6. Detail page:



7. click ‘Exit Points’:



Select ‘Move Inventory’ as Exit Point, you can configure source movement zone and destination movement zone when inventory moved for which this workflow to be fired, to simplify it we can configure both for ‘Any’ as the logic itself will check if this is for picking with catch weight scanned.

8. a snapshot from local test:

