# StoreMan Retrieval Assistant

## Create retrieval work list, requirements 8.3.1 – 8.3.7

When the retrieval assistant is given a new job, it allows the user to divide the work into sections:

### Box retrieval

**C\_retrieval\_job.status = new job (0); job type = box retrieval (2) or disposal (3)**

Find where the boxes are supposed to be:  
Select … from box\_name n, box\_store bs, c\_rack\_number r, c\_tank\_map m  
where n.box\_cid=bs.box\_cid and bs.rack\_cid=r.rack\_cid and r.tank\_cid=m.tank\_cid  
and bs.retrieval\_cid = jobID;

List the name, current structure and expected location of each box. The location should include site+location+name+layout, as it does in StoreMan’s storage browser

Allow the user to divide up the list (this may not be necessary; see below for more details). After division, show each section in turn and allow user to sort by location, otherwise show the whole list and allow them to sort that.

Ask user to save changes with the option of going back to re-order if necessary.

Insert an entry into c\_box\_retrieval for each box, recording the section it is in and its position within that section. Update c\_retrieval\_job set status=in progress (1)

### Sample retrieval

**C\_retrieval\_job.status = new job (0); job type = sample retrieval (4) or disposal (5)**

Find the cryovials to be retrieved:  
Select … from cryovial c, cryovial\_store cs, box\_name n, box\_store bs, c\_rack\_number r, c\_tank\_map m   
where c.cryovial\_id=cs.cryovial\_id and n.box\_cid=cs.box\_cid and n.box\_cid=bs.box\_cid and bs.status=6 and bs.rack\_cid=r.rack\_cid and r.tank\_cid=m.tank\_cid and cs.retrieval\_cid=jobID;

Roughly half of these will be for the primary aliquot (i.e. cryovial.aliquot\_type\_cid = c\_retrieval\_job.primary\_aliquot). The other half will be for the secondary aliquot. Primary and secondary should have the same barcode.

Entries for the *primary* aliquot could have a destination defined. You could find these using a left join (but not on a distributed database, hence no location):  
Select … from cryovial\_store s1 left join cryovial c on c.cryovial\_id=s1.cryovial\_id  
left join box\_name n1 on n1.box\_cid=s1.box\_cid  
left join cryovial\_store s2 on s1.cryovial\_id=s2.cryovial\_id and s2.status=0  
left join box\_name n2 on n2.box\_cid = s2.box\_cid  
where s1.retrieval\_cid = jobID

Create suitable box\_name and cryovial\_store entries if no destination boxes have been defined (but only for one cryovial in each pair)

Piece this information together to create a list giving the cryovial barcode, destination box, position, current box, position, structure and location of the primary and secondary

Display the size of the job and ask user if they want to divide up the list. If they do:

1. Ask them the maximum section size (default = 500 cryovials)
2. Calculate slot/box (where c\_box\_size.box\_size\_cid = box\_content.box\_size\_cid)
3. Ask them to select the size of first section from a list – it must be a multiple of the box size (from 2) and no more than the maximum (from 1)
4. Allocate the appropriate number of destination boxes to the first section
5. Repeat steps (2) and (3) until every entry has been allocated to a section

After division, show each section in turn and allow the user to sort by location, otherwise show the whole list and allow them to sort that.

Ask user to save changes with the option of going back to re-order if necessary.

Insert an entry into c\_box\_retrieval for each destination box, recording the section it is in and its position within that section (you might use position = 0 to indicate the section should be sorted by current sample location). Update c\_retrieval\_job set status=in progress (1)

### Other jobs

**C\_retrieval\_job.status = in progress (1) and job.type in (2,3,4,5):** act on list [tbd]

**C\_retrieval\_job.status not in (0,1) or job.type not in (2,3,4,5):** complain