# StoreMan Retrieval Assistant

Creating a retrieval work list (requirements 8.3.1 – 8.3.7) – updated 31 July 2013

See [K:/webcore/LIMS/database/v2.7/CentralDb.htm](file:///K:/webcore/LIMS/database/v2.7/CentralDb.htm) for database documentation

StoreMan allows the user to create lists of boxes or cryovials to be retrieved for analysis (8.2) or disposal (7.2). Sample retrieval for analysis may specify two aliquots: the secondary can be used if the primary is not available. The retrieval assistant (8.3) guides a user through the retrieval process. The first stage, when the retrieval assistant is given a new list to work on, is to create a retrieval plan divided up into manageable chunks.

### Box retrieval

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

Find where the boxes are currently stored:  
Select b.external\_name as box, s.external\_name as site, m.position, v.external\_full as vessel, m.shelf\_number, r.external\_name as rack, bs.slot\_position from box\_name b, box\_store bs, c\_rack\_number r, c\_tank\_map m, c\_object\_name s, c\_object\_name v where b.box\_cid=bs.box\_cid and bs.rack\_cid = r.rack\_cid and r.tank\_cid = m.tank\_cid and s.object\_cid = location\_cid and v.object\_cid = storage\_cid and bs.retrieval\_cid = :jobID;

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

Allow the user to divide up the list (if necessary; see below for more details). Show each chunk in turn and allow the user to sort by location, otherwise show the whole list and allow them to sort that.

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

Insert a record into c\_box\_retrieval for each box in turn and 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 samples to be retrieved:  
Select cryovial\_barcode, t.external\_name as aliquot, b.external\_name as box, cryovial\_position, s.external\_name as site, m.position, v.external\_full as vessel, shelf\_number, r.external\_name as rack, bs.slot\_position from cryovial c, cryovial\_store cs, box\_name b, box\_store bs, c\_rack\_number r, c\_tank\_map m, c\_object\_name s, c\_object\_name v, c\_object\_name t where c.cryovial\_id=cs.cryovial\_id and b.box\_cid = cs.box\_cid and b.box\_cid = bs.box\_cid and bs.status = 6 and t.object\_cid = aliquot\_type\_cid and bs.rack\_cid=r.rack\_cid and r.tank\_cid = m.tank\_cid and s.object\_cid = location\_cid and v.object\_cid = storage\_cid and cs.retrieval\_cid = :jobID;

At least half of these will be for the primary aliquot (i.e. cryovial.aliquot\_type\_cid = c\_retrieval\_job.primary\_aliquot). The others may be for the secondary aliquot. Primary and secondary tubes may have the same barcode but should always come from specimen entries with the same source name.

Entries for the primary aliquot *may* have a destination box defined. You could find these using a left join:  
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

If no destination boxes have been defined, ask for the box type and create suitable entries in box\_name.

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

The user may want to export and/or import the list to specify the retrieval plan – *this needs further thought.*

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 chunk size (default = 500 cryovials)
2. Calculate slots/box (where c\_box\_size.box\_size\_cid = box\_content.box\_size\_cid)
3. Ask them to select the size of first chunk 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 chunk
5. Repeat steps (2) and (3) until every entry has been allocated to a chunk

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

Ask the 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 chunk it is in, and a record into l\_cryovial\_retrieval for each cryovial, recording its position in the list. 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