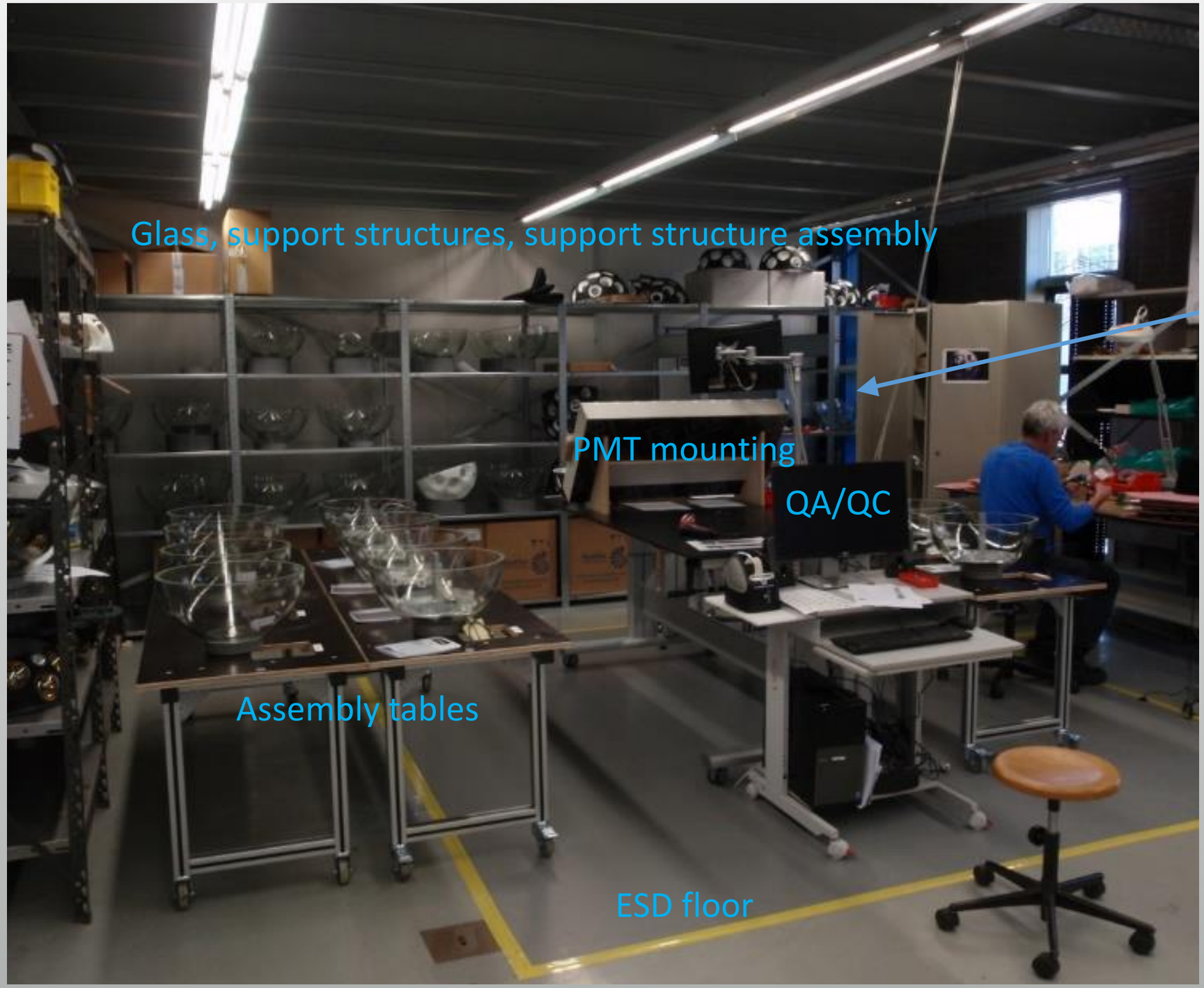
Product registration during DOM integration at Nikhef

Daan van Eijk
Fred Schimmel
Rene de Boer

Nikhef KM3NeT Group Meeting
March 10 2015


Glass, support structures, support structure assembly
PMT mounting
QA/QC
Assembly tables
ESD floor

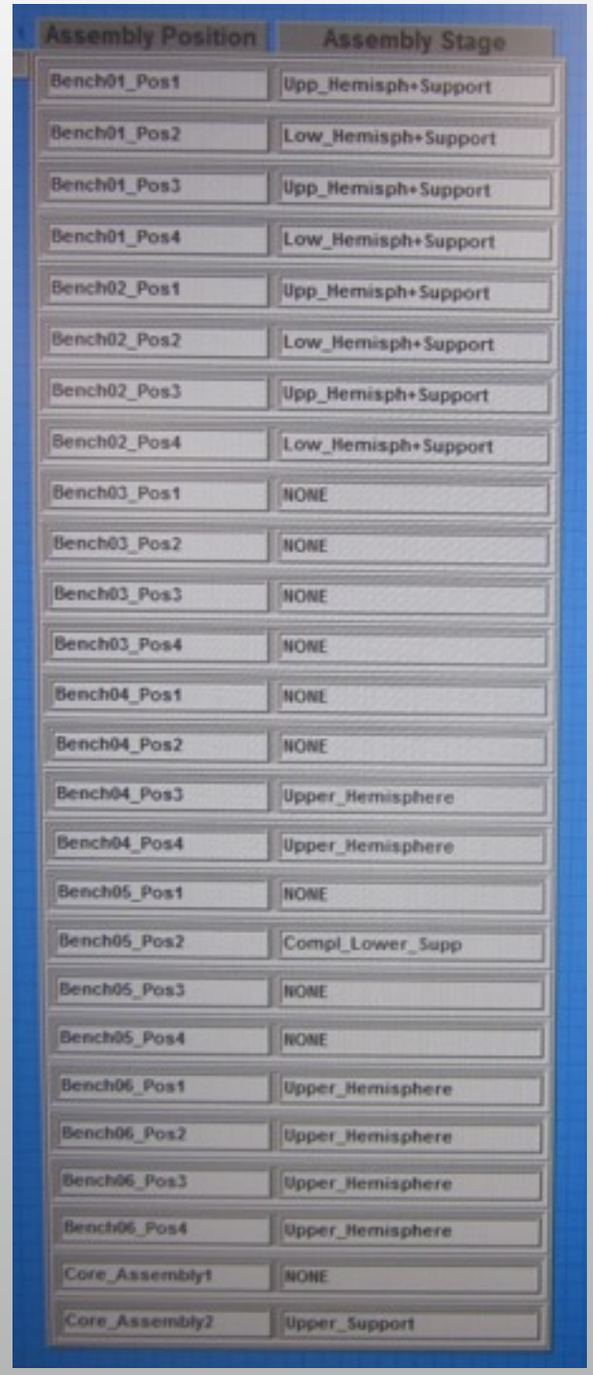
Why register products?

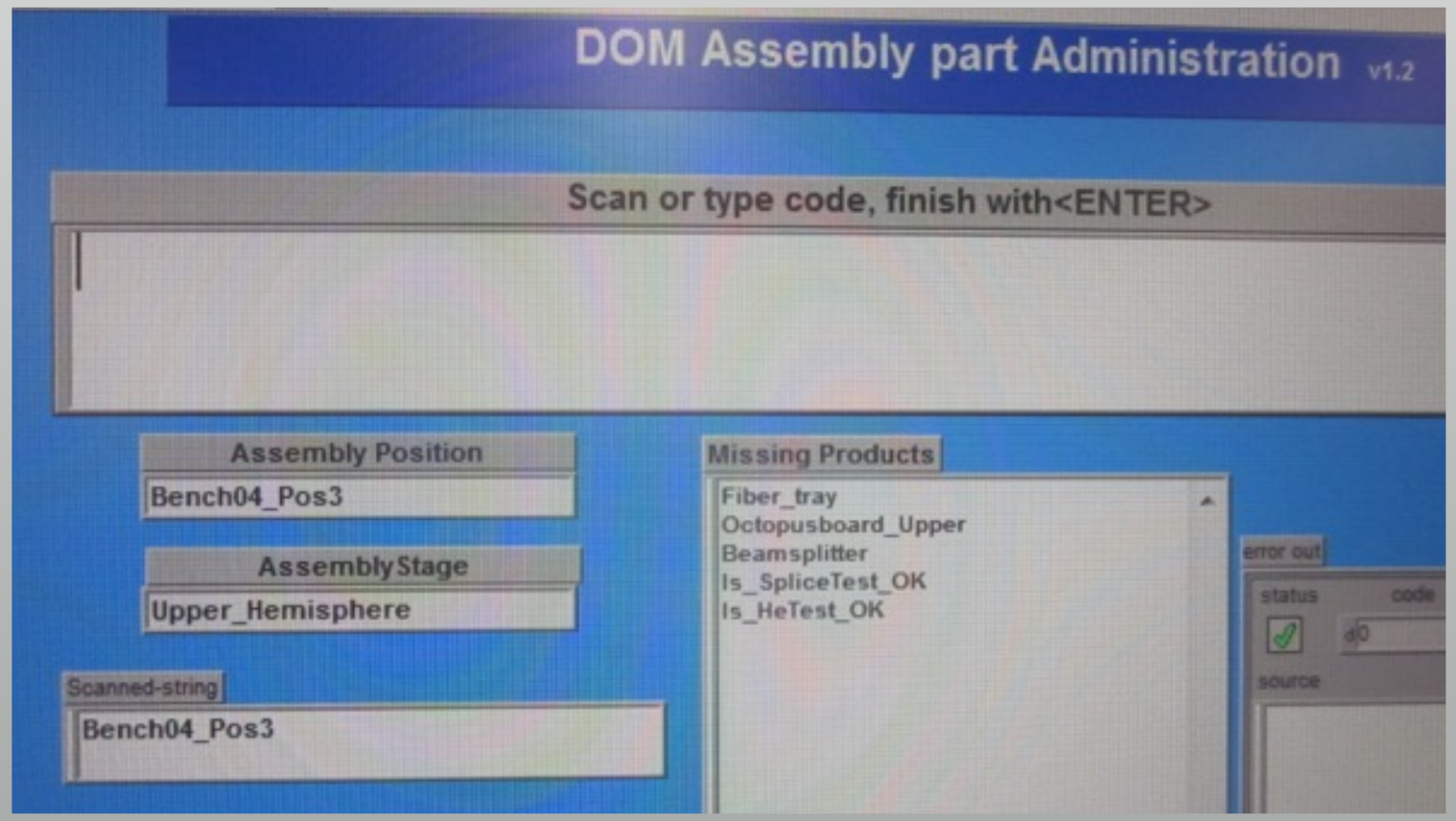
- Two goals:

1. **DAQ:**
    - Which CLB, piezo, nano beacon, Octopus boards?
    - Which PMT is where?
2. **QA/QC:**
    - To trace (batches of) products back if we ever find a problem with (certain) DOMs

Setup at Nikhef

- A PC with two LabView VI's running that reads from and writes to a (local!) Access database:
  - Overview program (reads current state of Bench Positions)
  - Product registration program (writes to DB)
- We make use of a QR-code scanner connected as keyboard input





Main objects

- The VI's work with the following main "objects"
- Bench Positions
- Assembly Stages
- Actions

Bench positions

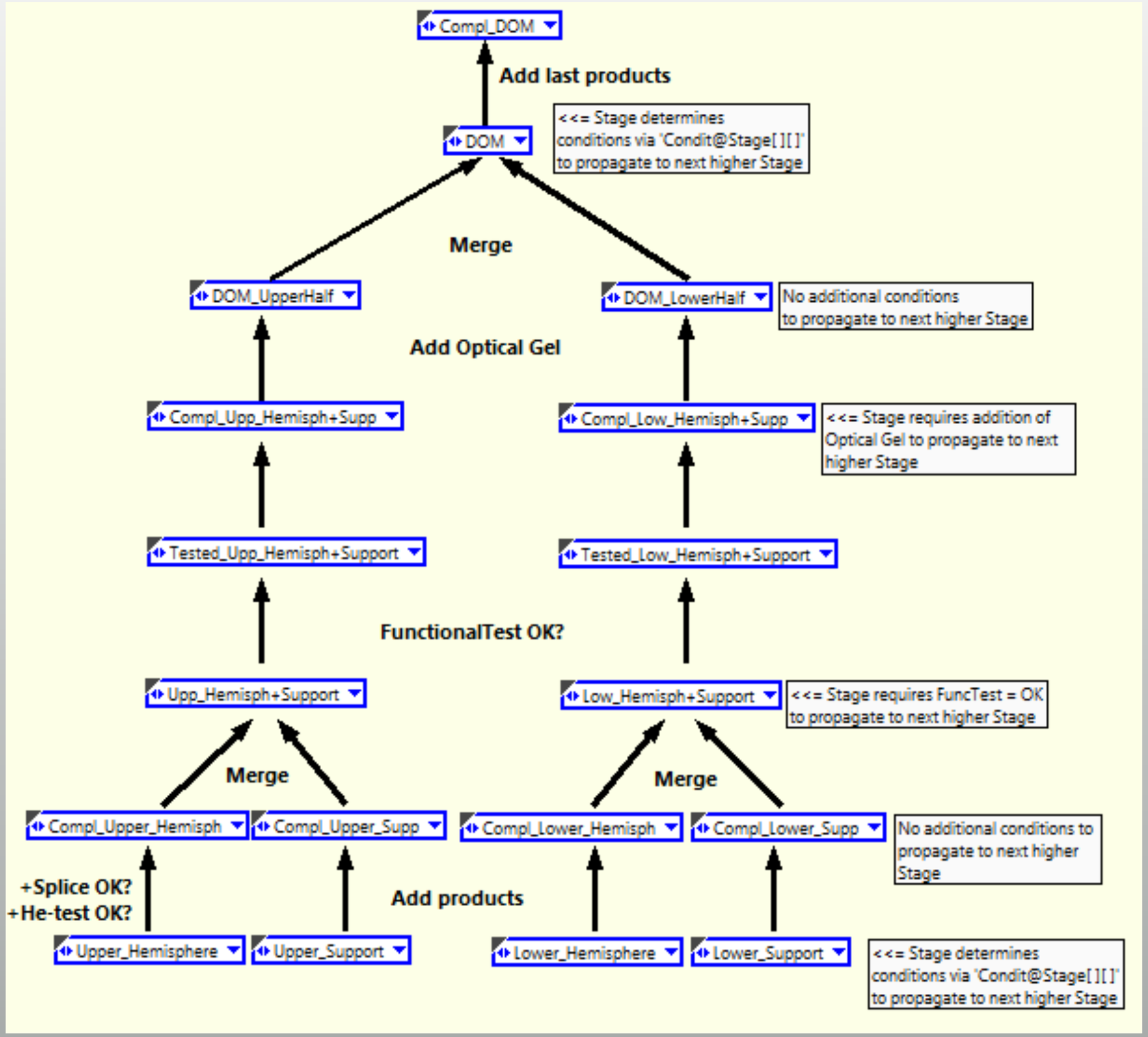
- Bench01_Pos1 - Bench01_Pos4

- …
- Bench06_Pos1 - Bench06_Pos4

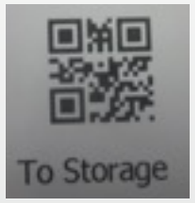
- Core_Assembly1 - Core_Assembly2

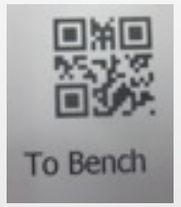
- Store01_Pos01 - Store01_Pos10

Assembly Stages



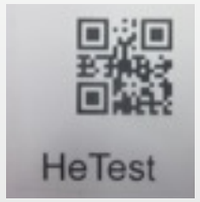
Actions

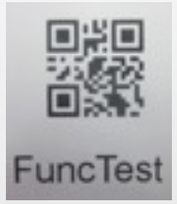


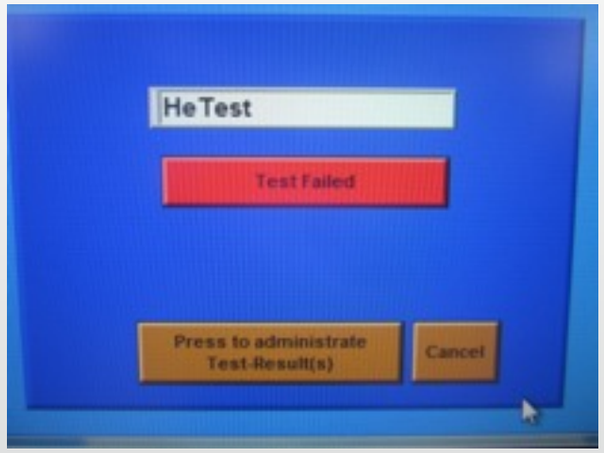


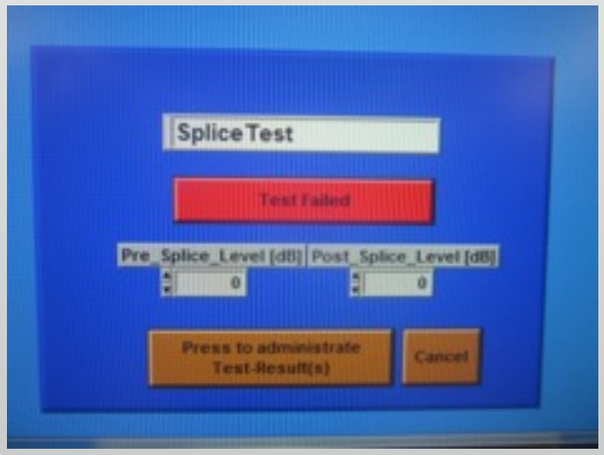
- **Merge**
  - Usage: scan <<Merge>>, scan <<1st bench pos>>, scan <<2nd bench pos>>
  - In database, it cuts the products in the row from <<2nd bench pos>> and pastes them to their corresponding columns in the row for <<1st bench pos>>, filling the ‘gaps’.
  - Check on Assembly Stages to see if Merge is allowed
  - <<2nd bench pos>> Assembly Stage changed to “None” —> position is free
- **To Storage**
  - Usage: scan <<bench pos>>, scan <<To Storage>>, scan <<store pos>>
  - In database, it cuts all products in the row from <<bench pos>> and pastes to <<store pos>>
  - <<bench pos>> Assembly Stage changed to “None” —> position is free
- **To Bench**
  - Other way around from “To Storage”
- **Replace**
  - Usage: scan <<Replace>>, program will ask what product needs replacement at this <<Bench Pos>>. Select the product. The program will ask for the new UPI.

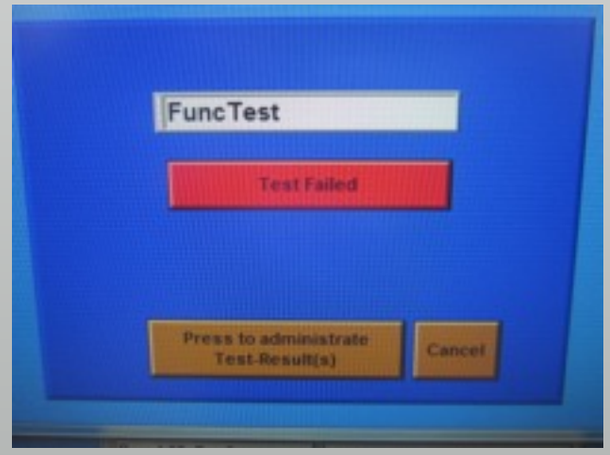
Actions: tests











- **He Test:**
  - Usage: scan <<He Test>>, click “Test Failed” to change the dialog to “Test Passed” and press <<administrate>>
- **Splice Test:**
  - Idem, but also enter optical attenuation before and after splice
- **Func Test:**
  - As He Test
  - Work to be done here. Still awaiting an official Test Procedure (Giorgio?). What do we want to register?
  - And is this the way to register, or do we want dedicated Functional Test sheets to be parsed to KM3NeT DB?

Database tables

- UPI administration
- Operator
- Timestamp

  Rows: a row for every bench position

  Columns: bench position, assembly stage, products, tests

- Finished DOM UPI
- Finished DOM operator
- Finished DOM timestamp

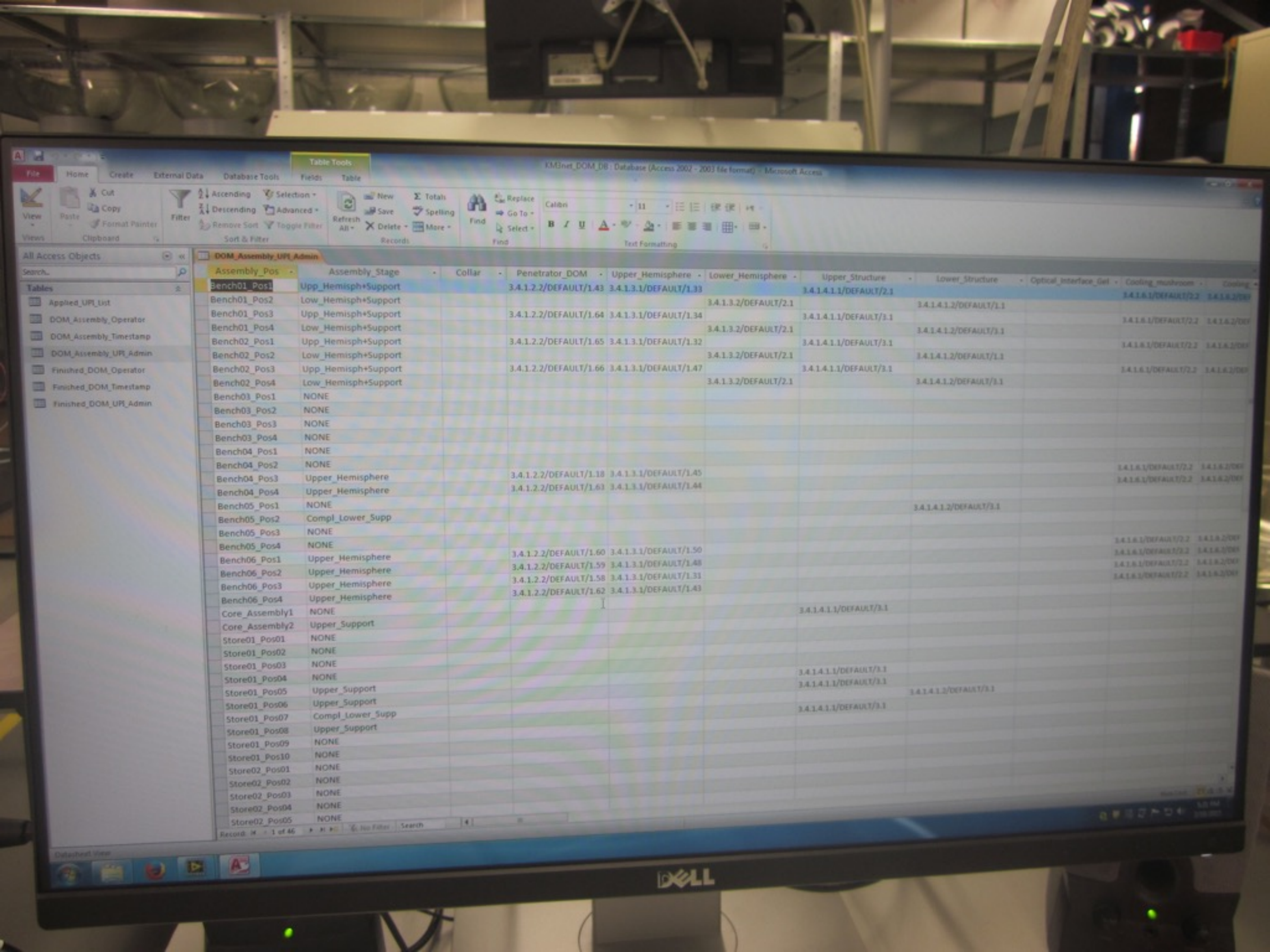
  Rows: a row for every completed DOM

  Columns: products, tests

- Applied UPI list

  Rows: a row for every unique product (no duplicates allowed!!!)

  Columns: UPI



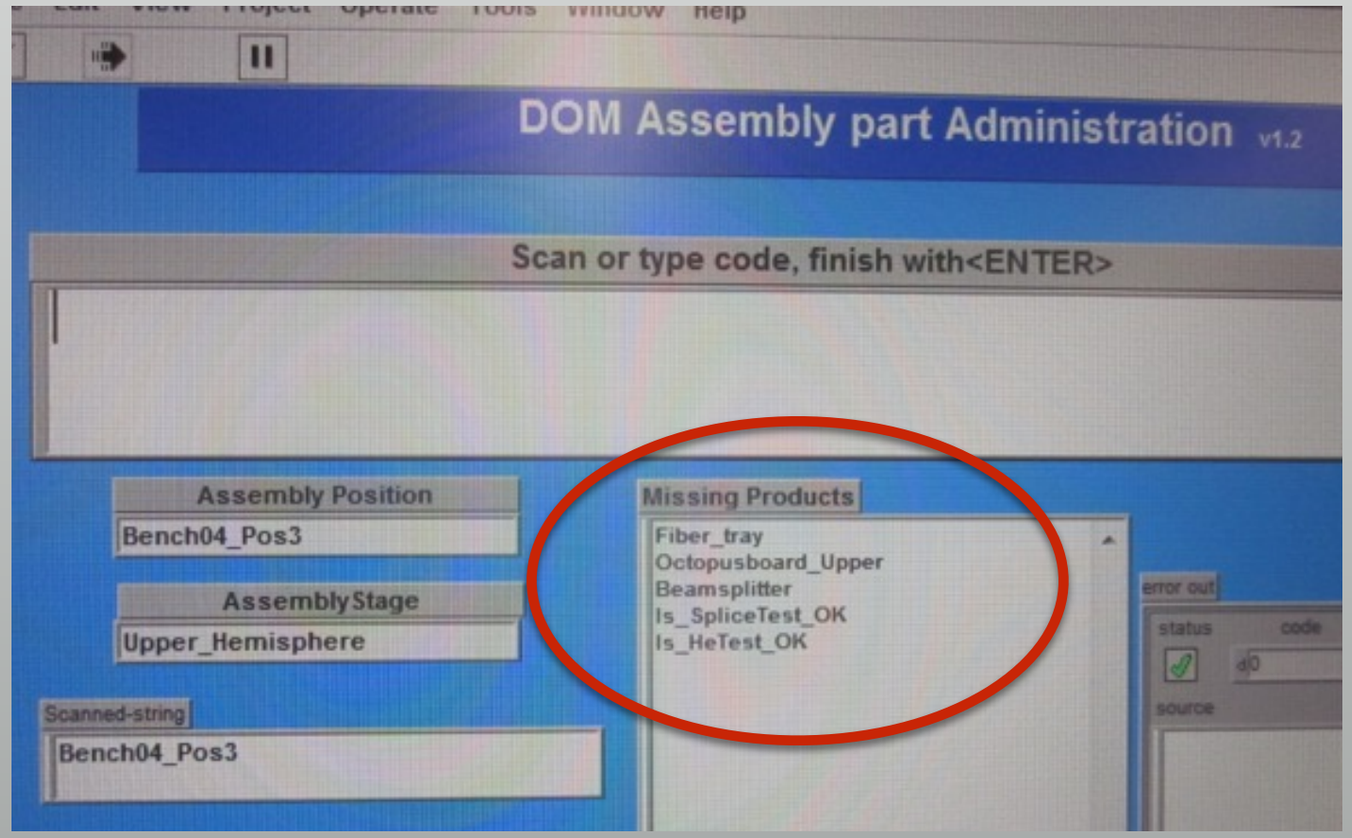
KMJnet_DOM_DB : Database (Access 2002 - 2003 file format) - Microsoft Access

DOM_Assembly_UPI_Admin

Assembly_Pos	Assembly_Stage	Collar	Penetrator_DOM	Upper_Hemisphere	Lower_Hemisphere	Upper_Structure	Lower_Structure	Optical_Interface_Gel	Cooling_mushroom	Cooling
Bench01_Pos1	Upp_Hemisph+Support		3.4.1.2.2/DEFAULT/1.43	3.4.1.3.1/DEFAULT/1.33		3.4.1.4.1.1/DEFAULT/2.1				
Bench01_Pos2	Low_Hemisph+Support				3.4.1.3.2/DEFAULT/2.1		3.4.1.4.1.2/DEFAULT/1.1			
Bench01_Pos3	Upp_Hemisph+Support		3.4.1.2.2/DEFAULT/1.64	3.4.1.3.1/DEFAULT/1.34		3.4.1.4.1.1/DEFAULT/3.1				
Bench01_Pos4	Low_Hemisph+Support				3.4.1.3.2/DEFAULT/2.1		3.4.1.4.1.2/DEFAULT/3.1			
Bench02_Pos1	Upp_Hemisph+Support		3.4.1.2.2/DEFAULT/1.65	3.4.1.3.1/DEFAULT/1.32		3.4.1.4.1.1/DEFAULT/3.1				
Bench02_Pos2	Low_Hemisph+Support				3.4.1.3.2/DEFAULT/2.1		3.4.1.4.1.2/DEFAULT/1.1			
Bench02_Pos3	Upp_Hemisph+Support		3.4.1.2.2/DEFAULT/1.66	3.4.1.3.1/DEFAULT/1.47		3.4.1.4.1.1/DEFAULT/3.1				
Bench02_Pos4	Low_Hemisph+Support				3.4.1.3.2/DEFAULT/2.1		3.4.1.4.1.2/DEFAULT/3.1			
Bench03_Pos1	NONE									
Bench03_Pos2	NONE									
Bench03_Pos3	NONE									
Bench03_Pos4	NONE									
Bench04_Pos1	NONE									
Bench04_Pos2	NONE									
Bench04_Pos3	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.18	3.4.1.3.1/DEFAULT/1.45						
Bench04_Pos4	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.63	3.4.1.3.1/DEFAULT/1.44						
Bench05_Pos1	NONE						3.4.1.4.1.2/DEFAULT/3.1			
Bench05_Pos2	Compl_Lower_Supp									
Bench05_Pos3	NONE									
Bench05_Pos4	NONE									
Bench06_Pos1	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.60	3.4.1.3.1/DEFAULT/1.50						
Bench06_Pos2	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.59	3.4.1.3.1/DEFAULT/1.48						
Bench06_Pos3	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.58	3.4.1.3.1/DEFAULT/1.31						
Bench06_Pos4	Upper_Hemisphere		3.4.1.2.2/DEFAULT/1.62	3.4.1.3.1/DEFAULT/1.43						
Core_Assembly1	NONE					3.4.1.4.1.1/DEFAULT/3.1				
Core_Assembly2	Upper_Support									
Store01_Pos01	NONE									
Store01_Pos02	NONE									
Store01_Pos03	NONE					3.4.1.4.1.1/DEFAULT/3.1				
Store01_Pos04	NONE					3.4.1.4.1.1/DEFAULT/3.1				
Store01_Pos05	Upper_Support						3.4.1.4.1.2/DEFAULT/3.1			
Store01_Pos06	Upper_Support					3.4.1.4.1.1/DEFAULT/3.1				
Store01_Pos07	Compl_Lower_Supp									
Store01_Pos08	Upper_Support									
Store01_Pos09	NONE									
Store01_Pos10	NONE									
Store02_Pos01	NONE									
Store02_Pos02	NONE									
Store02_Pos03	NONE									
Store02_Pos04	NONE									
Store02_Pos05	NONE									

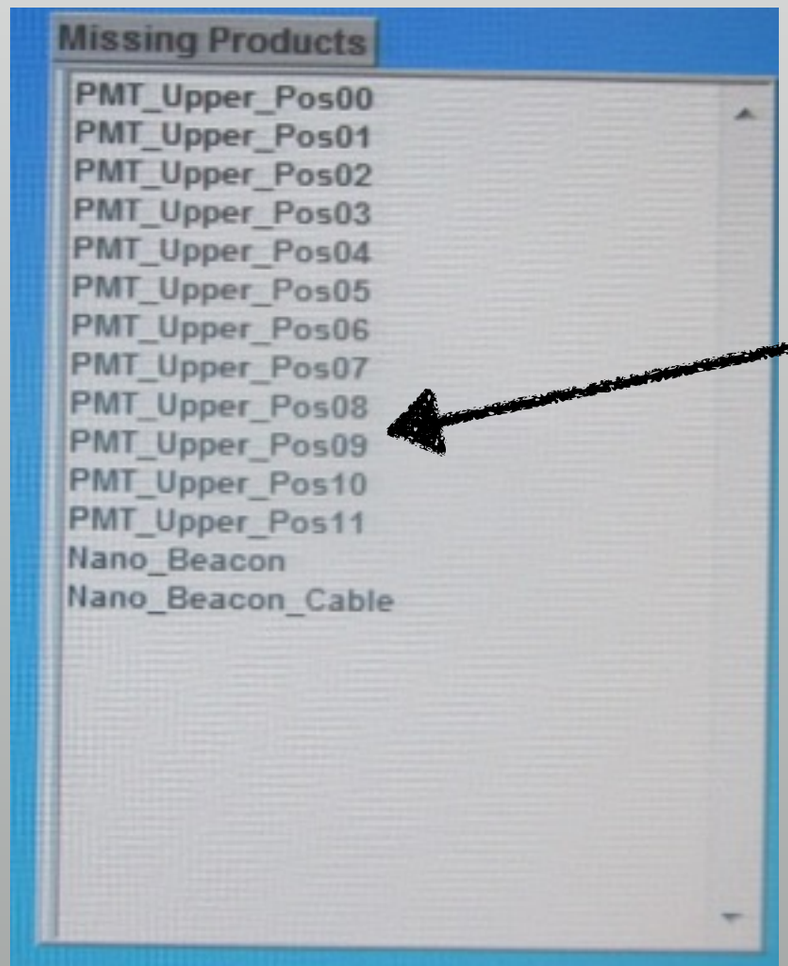
VI is integration checklist

- Every Assembly Stage demands a number of products and/or tests to advance to the next Assembly Stage
- This is represented in the Missing Products List
  - Note: A test is treated as a product in this case



Special case: PMT integration

- Integration of the PMTs is special:
  - We also need to record the position in the mechanical structure
- Whenever a PMT QR is scanned, the VI tells the user he/she should scan a position first, and only then scan the PMT QR.
- Only then will a PMT *position* disappear from the Missing Product List



Usage of local database

- Currently for DOM test setup (Linux system):
  - Python script to dump UPI table from Access to CSV file.
    - If someone has a better idea, please…
  - Read the CSV file, store rows as python lists
  - Do whatever you want:
    - Write to Excel/XML file (for upload of integration info to KM3NeT DB)
    - Get PMT HV settings from KM3NeT DB
      - See example at /home/dom0user/Documents/PMT_HV_settings/Get_PMT_HV_settings.py on kennet
    - …

For DOM testers

- Very important to note which DOM half you're picking up from which Bench Position

- You need that information to get correct PMT HV settings from KM3NeT DB

- Very important to put back the DOM half at its original position!

DOM identification

- When mounting an upper hemisphere on a Bench Position, we decide *beforehand* which DOM this upper hemisphere is going to be.

- Do we still want to keep a local count? This number is NOT used in our local DB nor in the KM3NeT DB

- Only used so far in the tests —> better to use UPI?

- Then how to identify lower halves?

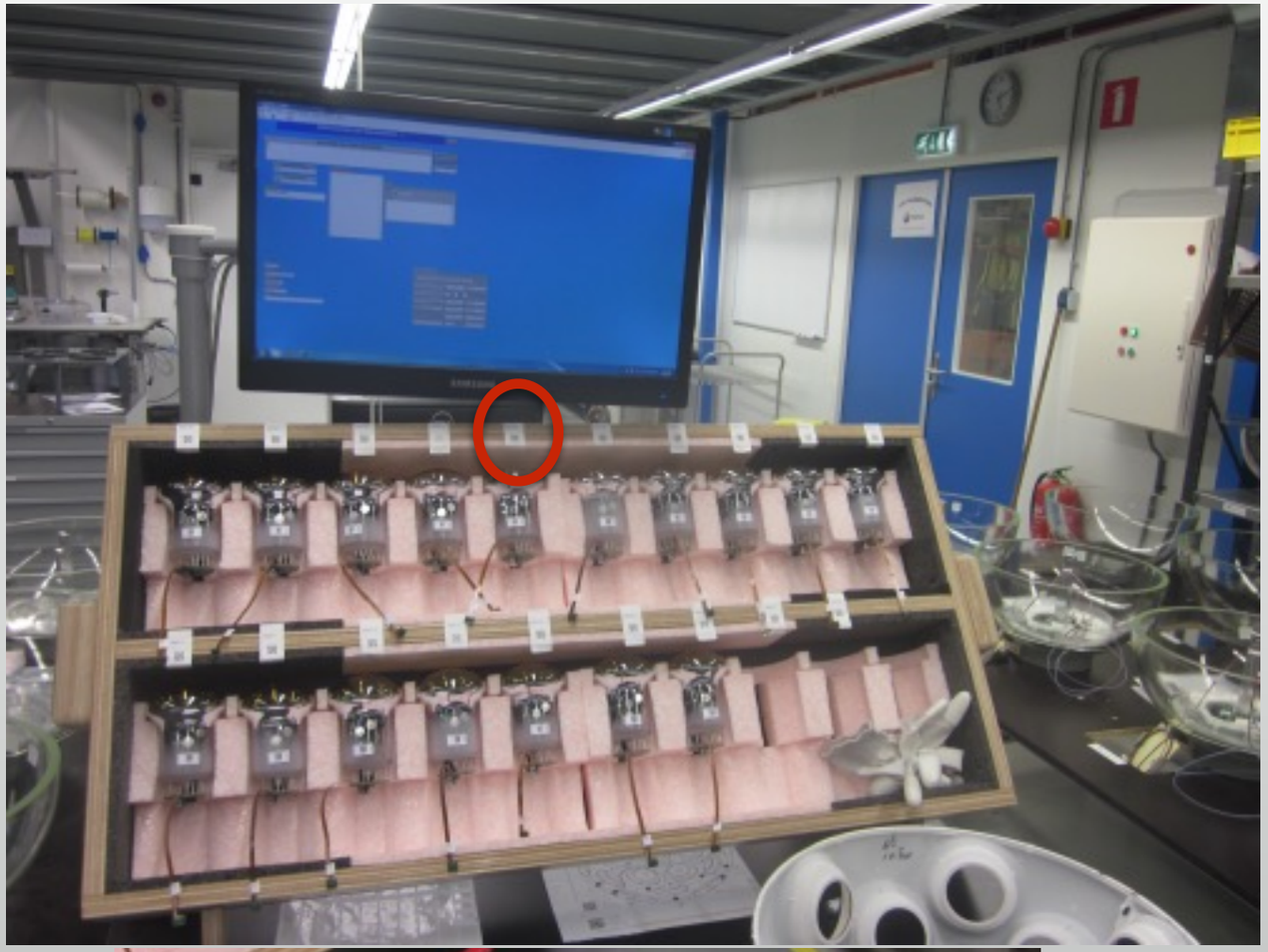
- Do we need keep halves together?

Integrate SFP channel 38C!!!

Conclusions

- At Nikhef, we use a Windows PC with a Tabview VI to register products during DOM integration
- Note which DOM (half) you pick up from which Bench Position before testing
- Be sure to put back the DOM (half) at the correct Bench Position after testing
- To do:
  - What and how do we want to register for tests?

PMT mounting



Slightly off-topic, but VERY useful and safe for PMT integration!