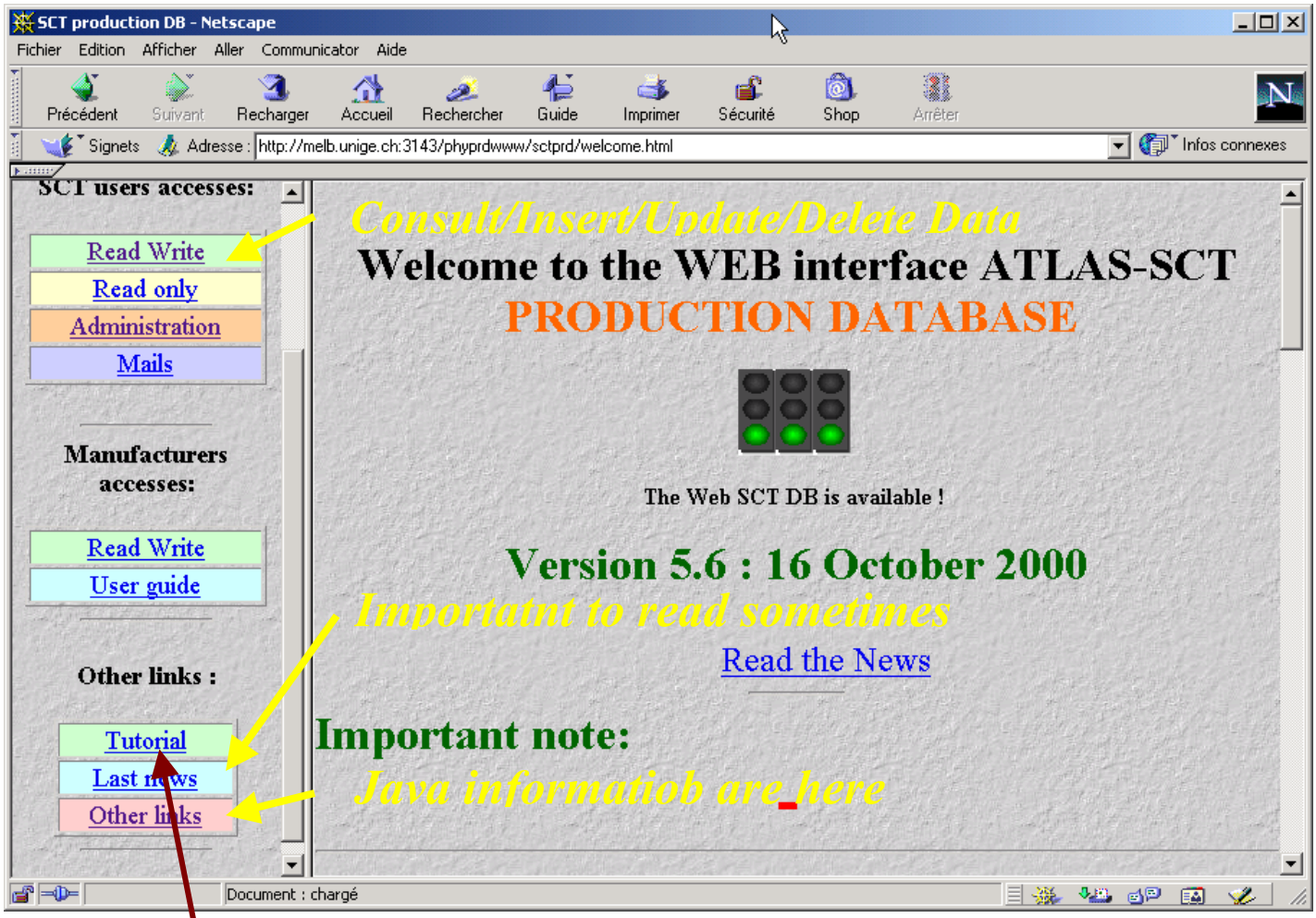
SCT Production DB
General tutorial

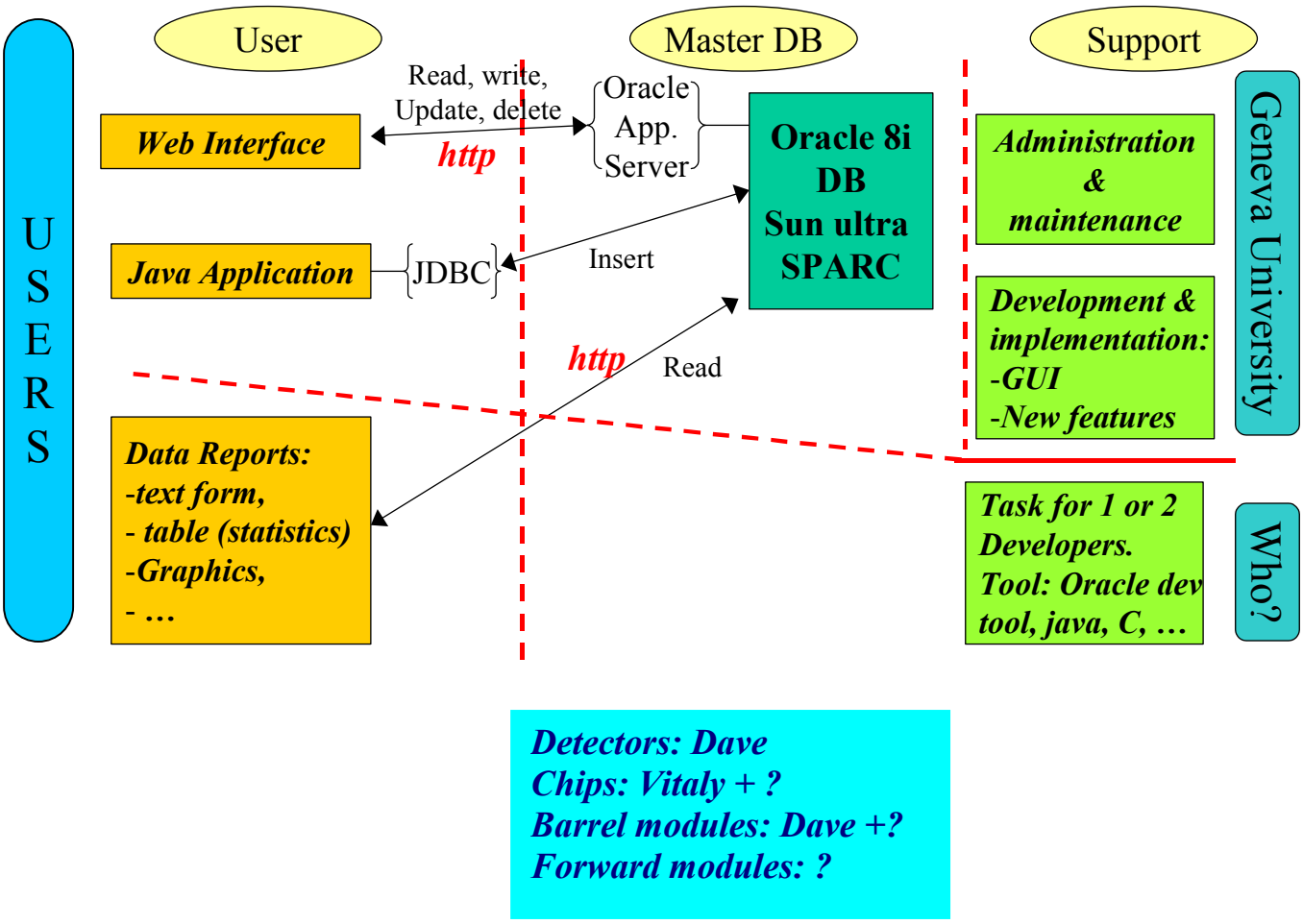
Production DB web page:
http://melb.unige.ch:3143/phyprdwww/sctprd/welcome.html



References:

- Atlas INDET note ATL-INDET-2002-015 (July 2002)
- Chip Tutorial (Feb 2002)
- Previous general tutorial (November 2001)

Data Base scheme



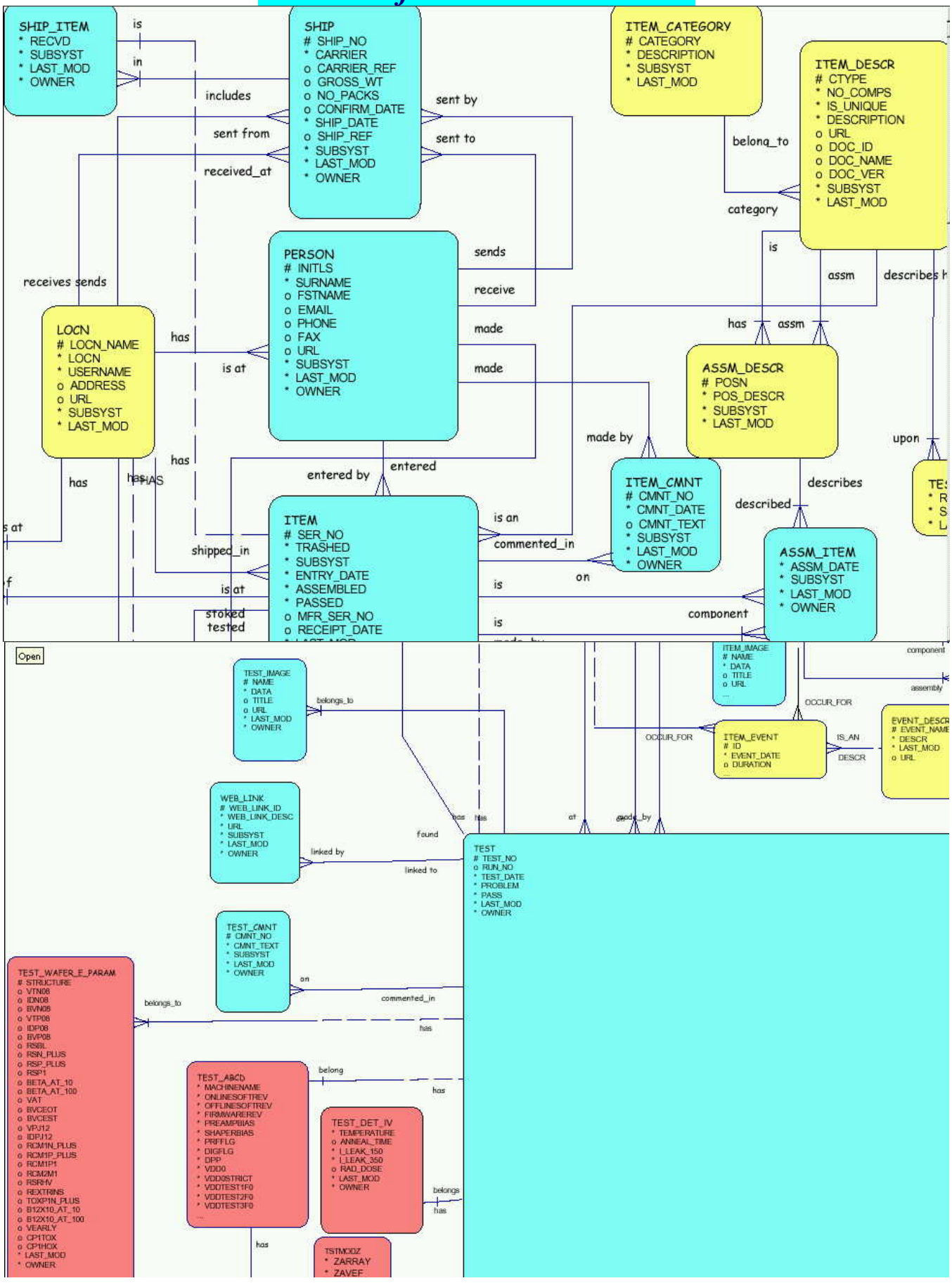
Important rule for users:

If you want to see data reports that is significant, you must enter the data and in a reliable way into the DB for:

- Items
- Shipment
- Assembly
- Events
- Tests

NB: Everyone must play the game with the existing rules otherwise it does not make any sense to use a DB.

Extracts of the Data Structure?



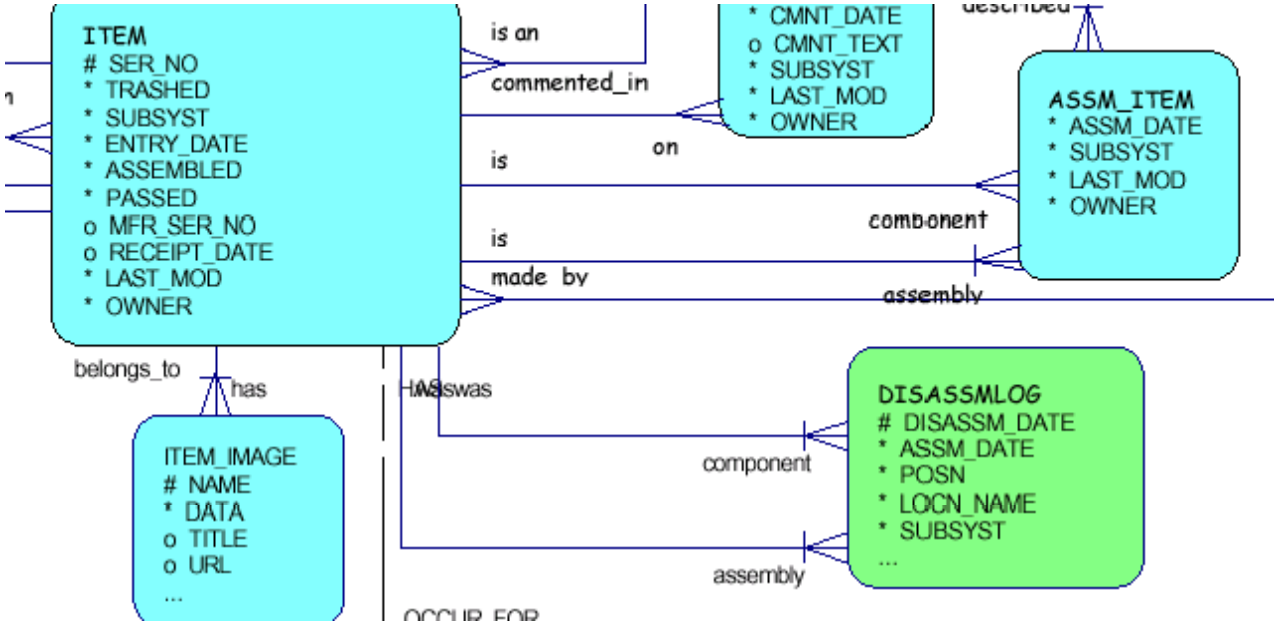
How it works?

This database model is based on a relational model and it has three major and general aspects:

- **Structures :** structures are well-defined objects (such as tables, views, indexes, and so on) that store or access the data of a database. The contained data can then be manipulated by operations.
- **Operations :** operations are clearly defined actions that allow the users to manipulate the data and the structures of a database. The operations on a database must adhere to a predefined set of integrity rules.
- **Integrity rules :** integrity rules are the laws that govern which operations are allowed on the data and the structures of a database. Integrity rules protect the data and the structures of a database.

In the case of the SCT database 3 classes of entity have been designed:

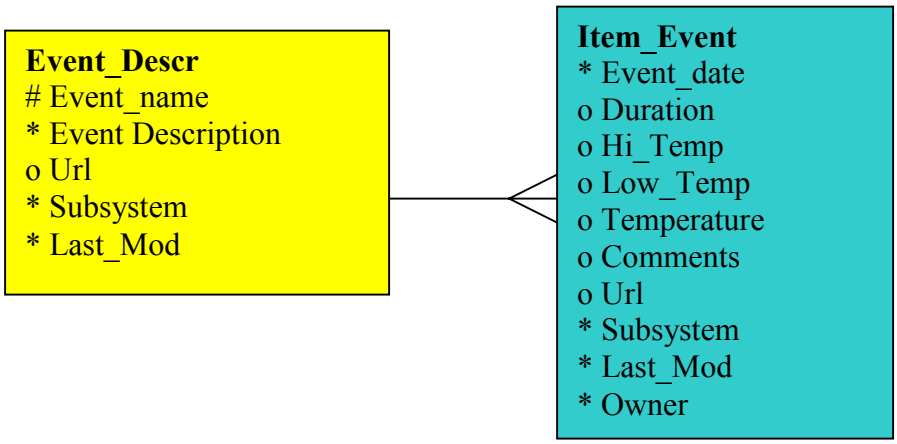
- **class 1 (in yellow):** is the definition structure for the description of items, tests, assembly and is not accessible to the standard user but by one or few administrators.
- **class 2 (In blue and red) :** is the user data structure that should hold the physical item, tests data. Individual users own their data.
- **class 3 (in green) :** is a history structure for assembly when a deletion is made like for a disassembly. The data are stored automatically in a log table when a deletion is triggered. The data in the disassembly history log table are then only readable by any authorized users.



Item Event

☞ Item Event table

A new entity has been implemented for defining any events that happened at a certain place and date. The access module is accessible under “Item” of the navigation panel.



The Event definition and description will have to be enriched times to times when there are needs identified ⇒ if definition are missing please contact me.

The actual description is:

Annealing
ChipDicing
p-Irradiation
n-Irradiation

The events (triggers)

An event or a set of events is always followed by an action on the DB like Insert, Update, Delete.

In the SCT DB, specific events exist for shipment, assembly / disassembly and they are triggered as followed:

- **Shipment validation:** Set the shipped items with ITEMS.OWNER = sent, ITEMS.LOCATION = unknown and SHIP_ITEMS.OWNER = Destination Location
- **Shipment reception:** Set the shipped items with ITEMS.OWNER = Destination OWNER, ITEMS.LOCATION = Destination Location
- **Assembly:** Set ITEMS.ASSEMBLED = YES
- **Disassembly:** Set ITEMS.ASSEMBLED = NO, ITEMS.LOCATION = Location of the Assembly, ITEMS.OWNER = OWNER of the assembly, fill ASSEMBLY_HISTORY

Recall : The serial number

All serial numbers in LHC project have 14 digits:

2	0	2	2	I	I	I	x	x	x	x	x	x	x

Where:

- 2022 : is the reserved number for SCT inside the ID of ATLAS project
- III : is the institute number that starts from 01 to 098
  099 and above is reserved for serial number generated by the system
- xxxxxxx : is the sequence number either **generated by the system** or a **free 7 digits** for each institute with its own institute number (using java).

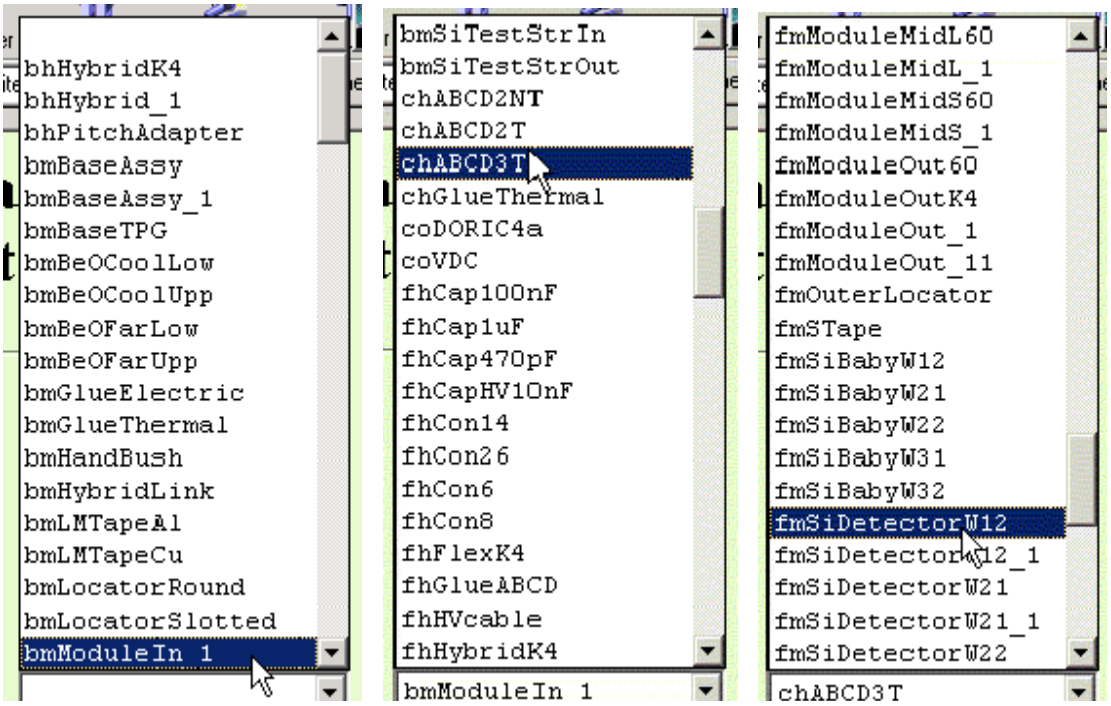
The definitions

The tables that contains definitions are:

- Item description
- Assembly description
- Test description
- Defect description
- Event description
- Item category
- Test list
- Test limits
- Manufacturers
- Location

The item type has a semantic in the name allowing the users to go directly to the type wanted:

- ‘bh’ : barrel hybrids
- ‘bm’ : barrel modules
- ‘ch’ : chips
- ‘fh’ : forward hybrids
- ‘fm’ : forward modules
- ‘oc’ : opto-components



How to get start with the web tools?

* *Web tools*

- **Just run a web browser (Netscape, Internet Explorer, …)**
- **Go to the web link:**
  http://melb.unige.ch:3143/phyprdwww/sctprd/welcome.html
- **Go into "SCT users accesses:" ☛Read Write**
- **Use your own Institute username/password**

Content of the categories in the navigation panel
(In Red: are read only access; In Green: are insert, consult, update and delete access)

Items	Assemblies	Tests	Others
Item description, Generic items, Chips, Detectors, Shipment , Basic reports.	Assembly description, Assembly, Tree reports, History.	All test description, (Test, defect, list) All detector test access module, XYZ module survey, chips, Optolinks, Low mass power tapes, Basic reports.	Location, Manufacturer, Person.

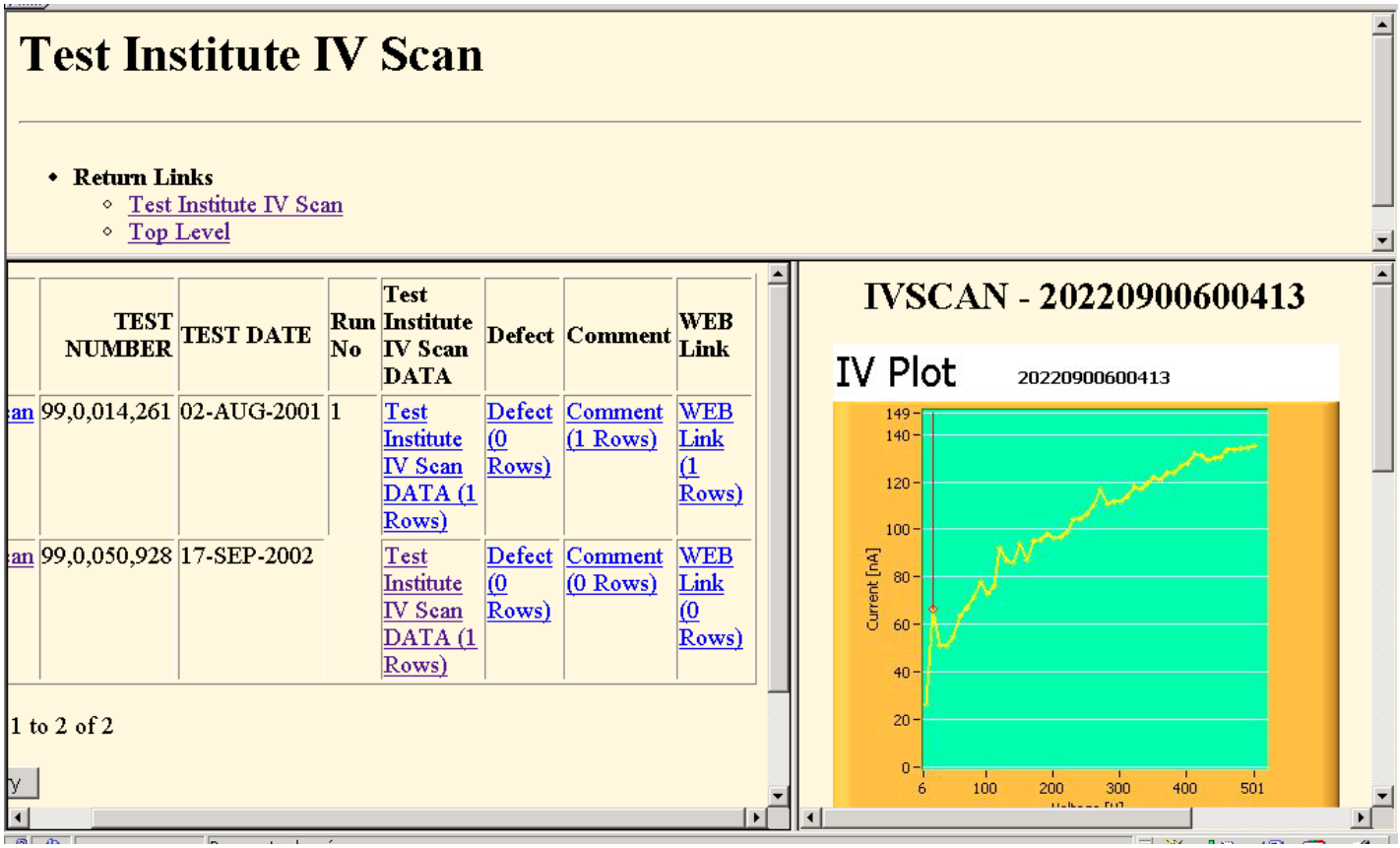
Web tools and functionalities

Any data user insertion is tagged with OWNER and LAST_MOD.
Note that OWNER ≡ USERNAME (used for the access web or java)

- **Direct data access via web modules** (read, insert, update, delete)

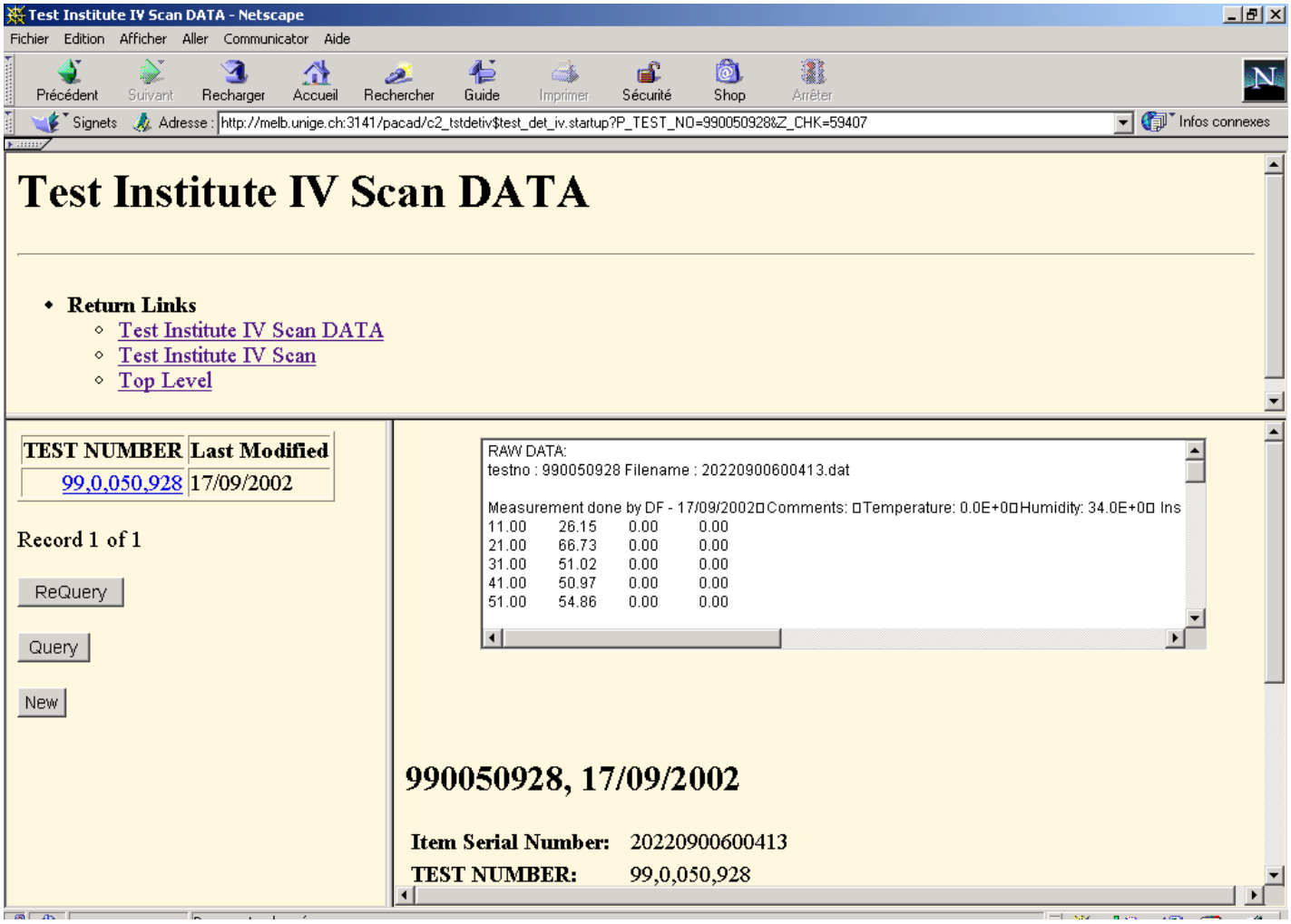
Any web browser (Netscape, Internet Explorer, …) must be able to run the specific access module of the navigation panel. The access of the data via the web module follows the tree of the data structure. Few Basic Reports are also available and allow summarizing some data in a table according to a specific query.
NB: The key in any of the access module is the Serial number of the items for which you want to consult the data.

The specificity of the web stays the interactivity links and image display (GIF or JPG) that have been register into the specific place of the DB.
For reading the content of uploaded raw data via the web, a java applet is started and a special frame with scroll bar is displayed (see illustration below).



Web tools and functionalities

Raw Data Display:



Web tools and functionalities

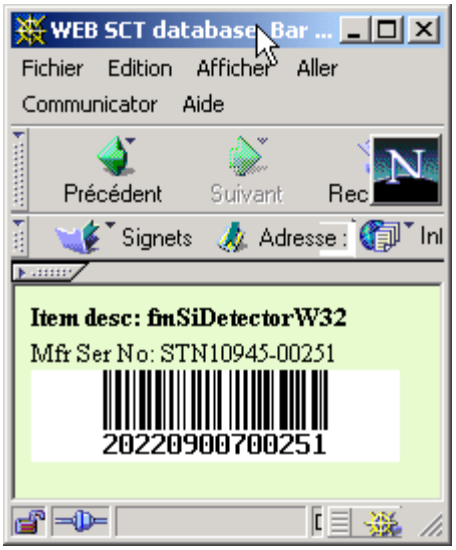
- **Bar code generation**

By entering the serial number of your item, the DB will check the validity of it and will generate a bar code display (type 128). This bar code can then be printed on a standard printer or a special one for small sticky papers.

They are 2 special cases:

- Barrel module assembly: it will generate the bar code of the module and the hybrid.
- Forward module assembly: it will generate the bar code of the module, the hybrid and the flex.

Normal item



Forward module

NB: A range of bar code can also be displayed/printed with a layout that can vary in width and height! A maximum of 200 is allowed!

Web tools and functionalities

- **Chip to hybrid/module assembly**

This operation consists of disassembling a chip from a gel pack assembly and then to assemble it to the hybrid or module.

Assembly Serial Number:	20220990004161	(Mandatory)
Gelpack Serial Number:	20220990304145	(Mandatory)
Assembly Position #	**Gelpack Position # (Optional)**	**Assembly Date (Format: DD Mon YYYY)**
1	10	Today
2	11	Today
3		Today
4		Today
5		Today
6		Today
7		Today
8		Today
9		Today
10		Today
11		Today
12		Today

Submit Reset

SCT Database - Chip Disassembly/Assembly Action

The following records are under deletion (This action is equivalent to remove chips to gelpack):

Gelpack Ser No	Chip Ser No	Position #	Assembly status
20220990304145	2C220990304151	10	Deleted
20220990304145	2C220990304152	11	Deleted

The chip to hybrid assembly records are:

Gelpack pos #	Assembly Ser No	Chip Ser No	Hybrid position #	Assembly date	Status
10	20220990004161	20220990304151	1	18-FEB-02	Assembled
11	20220990004161	20220990304152	2	18-FEB-02	Assembled

SUCCESS !!!

*Action made by **W3SCTDEV***
18-02-2002 12:51:02

Web tools and functionalities

- **Web basic reports**

The web basic reports allow the users to access widely the data from various tables. The list of available web reports is:

Description	***Found under***
Item basic report	Item
Detector delivery status	Item
Shipped Item	Item
Assembly description tree report	Assembly
Assembly tree report	Assembly
All Tests Report	Test
Institute IV report	Test
Defect report	Test

Example of Shipped item report:

SCT Database - Shipped Item report

Item info for Ser_no: 20220900600413

Type	Location now	Assembled	Test passed status
fmSiDetectorW31	Geneva	YES	All required tests performed and OK!

Shipment info for item # : 20220900600413

Shipment date	Shipment #	From	Person	To	Person	Carrier	Sent?	Received?	
17-APR-01	990000590	Hamamatsu	HPK	Geneva	DF	YUSEN AIR CARGO	YES	YES	○
15-JUN-01	990000699	Geneva	AGC	Moscow, NPI	MM	CERN DHL	YES	YES	○
31-AUG-01	990000909	Moscow, NPI	NB	CERN	AGC	DHL	YES	YES	○
20-OCT-01	990000910	CERN	RF	Geneva	DF	Shipment from Moscow	YES	YES	○

To get information about the list of items that was into a same shipment box, please select above and Submit!

Submit Reset

Web tools and functionalities

- **Display of test status**

The TEST.Pass [Boolean] attribute is the key for displaying the test status of an item!
Basically this function scans all the mandatory and optional tests and gives the result in a string with red, green or black color. The results are such as:

- "All required tests performed and OK!"
- "All required tests not yet performed, and OK!"
- "All required tests performed and failed!"
- "All required tests performed and OK! Optional Test performed Failed!"
- "No test list defined!"

NB: Only every last specific tests are considered!

How to get start with the java tools?

** Java tools*

- **Be sure that at least JRE (Java Run time Environment) is installed (otherwise install it following the the SCT DB web instruction)**
- **Be sure that Classes111.zip (for java version 1.1.x) or Classes12.zip (for java vesrion 1.2.) is downloaded locally on your machine**
- **Be sure that the application jar files are also downloaded**
- **Set the above application into your CLASSPATH environment**
- **Once it is done, it is forever unless a new upgrade is needed**
- **Run manually or automatically into a DAQ software system the command: java UploadXxxxData … or another application.**
- **Control that the upload succeeded either manually or automatically.**

How to set the CLASSPATH on linux (cshell):

setenv CLASSPATH /myclassdirectory/classes111.zip
setenv CLASSPATH
${CLASSPATH}:/myclassdirectory/SCTTEST_3_15_P.jar
setenv CLASSPATH
${CLASSPATH}:/myclassdirectory/SCTABCDCHIPS_3_01_P.jar
setenv CLASSPATH
${CLASSPATH}:/myclassdirectory/SCTAssm.jar
setenv CLASSPATH
${CLASSPATH}:/myclassdirectory/ SCTspecialDev.jar

How to set the CLASSPATH on other machines:

Refer to "Tutorial" link into the navigation panel

Java tools versus operations

Upload operation	***Tool to use***
Initial registration	***java UploadItemData File SCTusername SCTpassword***
Item image	***java Upload_ItemImage SCTusername SCTpassword ItemSerialnumber imageFilename [-T Title] [-U url]***
Shipment	***java UploadShipData File SCTusername SCTpassword***
Shipment validation	***Web*** *(by Updating "Confirmation Date")*
Shipment rececpt.	***Web*** *(by receiving the items)*
Assembly	***Java UploadAssmData File SCTusername SCTpassword***
Tests (any)	***java UploadTestData File SCTusername SCTpassword***
Test image	***java Upload_ItemImage SCTusername SCTpassword testnumber imageFilename [-T Title] [-U url]***
ABCD chips and wafer registration	***java UploadABCDChips FileName SCTusername SCTpassword***

The other specific java tools considered as special development:

Operation	***Tool to use***
Download last raw data of a particular test	***java GetLastRawData SerNo testname outfilename UserReadAccess***
Download raw data	***java DownloadRawData testnumber outfilename SCTusername SCTpassword OutputFilename***
Get the corresponding chip position versus serial number	***java GetChipPosSerNo WaferMfrSerNo ChipType OutputFile UserReadAccess***

What can be already uploaded with Java tools?

Tests:

- Institute IV scan (detector current measurement)
- Institute Module IV scan (module current measurement)
- Si detector depletion voltage test
- Si detector leakage current stability
- Si detector interstrip capacitance
- Si detector polysilicon bias resistance
- Si detector strip metal series resistance
- Si detector coupling capacitance
- Si detector implant sheet resistance
- Si detector flat band voltage
- Si detector Defects
- Si detector Visual Inspection
- Module Survey XY
- Module Survey Z
- Low Mass Tape
- ABCD wafer screening tests
- Wafer test parameters

Others:

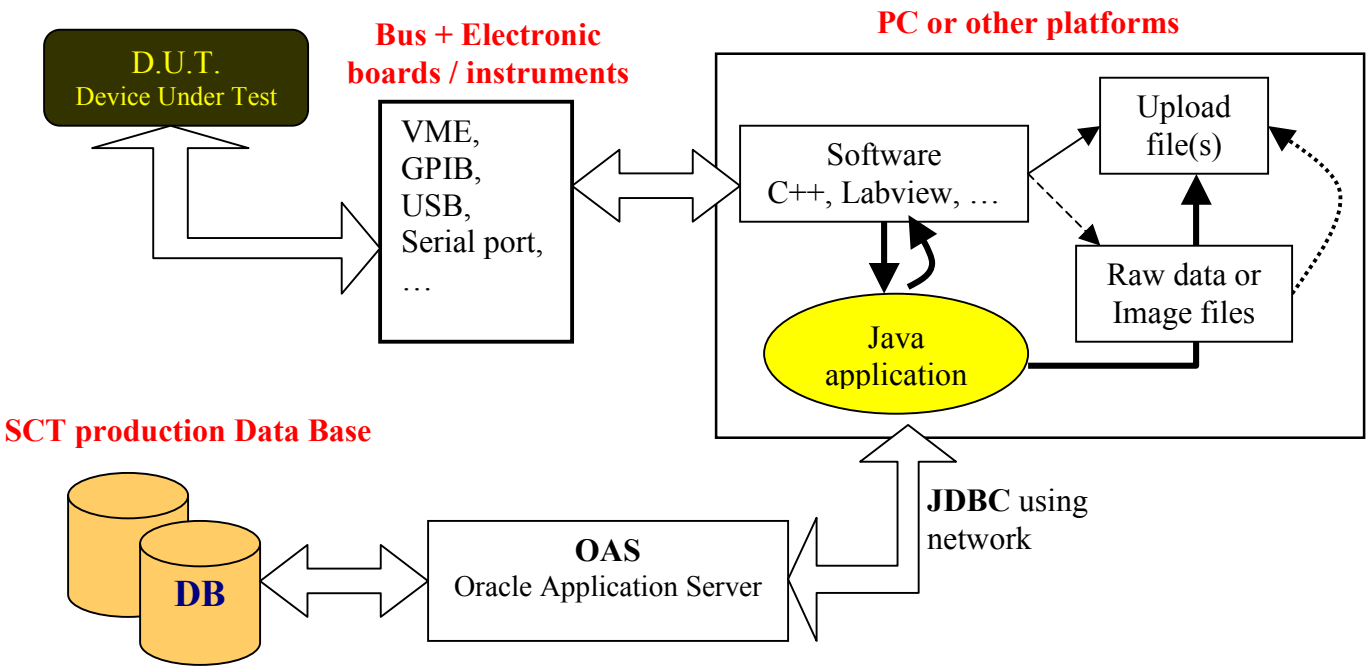
- Initial registration of items (with 7 digits freedom for the serial number)
- Shipment of items
- Assembly
- Item images
- Test images
- Initial registration of chips

Why Java applications?

It is useful for:

- ❖ Automated acquisition system
- ❖ Upload of mass data
- ❖ Registration of your own SerNo
- ❖ Practical for shipment of a lot of items
- ❖ Downloading raw data

General scheme of a test system interfaced with the SCT production DB



Initial registration of items

Command line:

java UploadItemData CommandFileName SCTusername SCTpassword

Inside the file:

```
# Example of datafile
# ====================
%Item
Serno          : 20220020000005
ctype          : bmGlueElectric
EDate          : 30/09/2002
ASSM           : YES
PASS           : NO
Initls         : PHD
LocnName       : Geneva
Mfr            : Geneva
MSerno         : MFRSERNO - 001
RDate          : 20/06/2002
%Item
SERNO          : 20220020000006
CTYPE          : bmGlueElectric
EDate          : 30/09/2002
ASSM           : YES
PASS           : NO
Initls         : PHD
LocnName       : Geneva
Mfr            : Geneva
MSerno         : MFRSERNO - 002
RDate          : 20/06/2002
```

Upload a shipment into the DB

Command line:

java GetChipPosSerNo Tutorial-Feb-2002 chABCD3T out.txt sctread
java UploadShipData CommandFileName SCTusername SCTpassword

Inside the CommandFileName (This is only an example) :

```
%Shipment
SHIP DATE : 03/05/2001
SENDER PERSON : DF
SENDER LOCATION : Geneva
DEST PERSON : SR
DEST LOCATION : CERN
CARRIER NAME : Private
Carrier Ref : No ref
References : No ref
NO Packs : 3
Weight : 12
#
#
%ShipItem
20220900600001
20220900600003
20220900600004
```

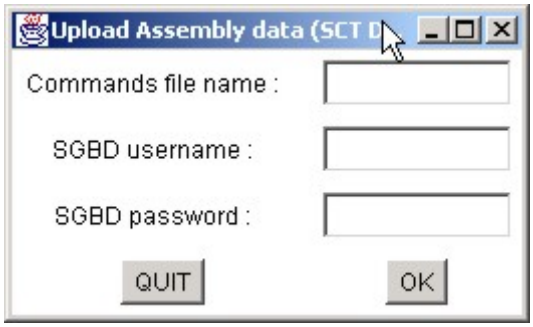
Optional fields.
If they are empty,
please remove the line!

Java tool to upload an assembly (chip to gelpack)

This application was originally developed for the chip to gelpack assembly but can also be used for any other authorized assembly.

The command line is:

Java UploadAssmData [CommandFileName SCTusername SCTpassword]



The format file is the following:

```
%Assembly
ASSEMBLY ITEM: 20220990000826
# CompSerNo     Position       Date
(DD/MM/YYYY)
20220990000002  1              28/09/2001
20220990000001  2              28/09/2001
20220990000010  3              28/09/2001
20220990000011  4              28/09/2001
```

NB: A new attribute "Assembly_date" is now mandatory for the assembly records.

Test Upload file rules

Examples of table and attributes specs:

Table: Test_xxx	O/M	Type	Max Range*
Attribute1	M	String	32 chars
Attribute2	O	String	60 chars
Attribute3	M	Boolean	YES/NO
Attribute4	M	Float	[0.5;6.5]
Attribute5	O	Byte	[0;128]

** if a value is outside the range the DB will refuse the record.*

Example of file to provide for the Upload according to specs above:

```
%NewTest
SERIAL NUMBER       : 20220990000019
TEST MADE BY        : DF
LOCATION NAME       : Geneva
Run number          : Here is my run number
TEST DATE           : 09/10/2001
PASSED              : YES
PROBLEM             : NO
#  Here starts the attributes for the new test that have been specified
%Test_xxx
# Attribute1                    Attribute2          Atribute3
"This is an example"  <blank(s)>  .  <blank(s)>    4.15
# Attribute4                    Attribute5
    5.1               <blank(s)>    .
%Comment
COMMENT             : Here is my test comment 1
%Web link
DESCRIPTION         : Here is my Web link description V2
URL             : http://www.cern.ch
%TEST Rawdata
FILENAME            : RawDataFile.txt
%TEST Image
FILENAME            : DataFile.txt
Url             : http://www.cern.ch
Title           : The title of the image
```

Mandatory part for any tets

Relative to the new test table: Test_XXX**

Optional part that can be add to any test

NB: Only the raw data section in the optional part can appear only once.
There is no limitation for: Web_link, test_comment and test_image
"#" can be used at any place at the beginning of each line for specifying a comment line
** In this specific section the order of the attributes must be respected.

Test Upload file rules

Another type of example (old fashion format):

```
%NewTest
SERIAL NUMBER    : 20220990000019
TEST MADE BY     : PhD
LOCATION NAME    : Geneva
Run number       : Here is my run number zp
TEST DATE        : 20/06/2000
PASSED           : YES
PROBLEM          : NO
# --
%ModSurveyXY
MHX              : -55.7
MHY              : 3.7
MSX              : 23.87
MSY              : -45.4
MIDXF            : 0.4
MIDYF            : 0.6
sepf             : 52.31
sepb             : 67.3
STEREO           : -22.65
a1               : 0.5
a2               : 0.3
a3               : 0.23
a4               : 0.1
MACHINE          : Name of the measurement machine
TEMPERATURE      : 23.5
# --
%Comment
COMMENT          : Here is my test comment 1
%Comment
COMMENT          : Here is my test comment 2
#
#
%Web link
DESCRIPTION      : Here is my Web link description V2
URL              : http://www.cern.ch
#
%TEST Rawdata
FILENAME         : DataFile.txt
#
%TEST Image
FILENAME         : DataFile.txt
Url              : http://www.cern.ch
Title            : The title of the image
```

Upload file – Example for wafer parameters

WAFER_E_PARAM unit	Opt/Man	type	error range	warning range
Structure	M	BYTE	[1,5]	[1,5]
VTN08	O	FLOAT	[0.0,5.0]	[0.730,1.130]
IDN08	O	FLOAT	[0.0,20.0]	[6.780,10.290]
BVN08	O	FLOAT	[0.0,25.0]	[8.000,17.780]
VTP08	O	FLOAT	[0.0,5.0]	[0.600,1.000]
IDP08	O	FLOAT	[0.0,20.0]	[3.850,5.900]
BVP08	O	FLOAT	[0.0,20.0]	[10.0,16.170]
RSBL	O	FLOAT	[0.0,1000.0]	[405.0,635.0]
RSN_PLUS	O	FLOAT	[0.0,200.0]	[48.0,72.0]
RSP_PLUS	O	FLOAT	[0.0,200.0]	[90.0,146.0]
RSP1	O	FLOAT	[0.0,5.0]	[2.000,2.700]
BETA_AT_10	O	FLOAT	[0.0,1000.0]	[120.0,350.0]
BETA_AT_100	O	FLOAT	[0.0,1000.0]	[120.0,350.0]
VAT	O	FLOAT	[0.0,500.0]	[80.0,190.0]
BVCEOT	O	FLOAT	[0.0,20.0]	[5.5,15.0]
BVCEST	O	FLOAT	[0.0,30.0]	[8.0,18.050]
VPJ12	O	FLOAT	[0.0,5.0]	[8.0,1.8]
IDPJ12	O	FLOAT	[0.0,50.0]	[7.0,25.0]
RCM1N_PLUS	O	FLOAT	[0.0,100.0]	[0.0,30.0]
RCM1P_PLUS	O	FLOAT	[0.0,200.0]	[0.0,80.0]
RCM1P1	O	FLOAT	[0.0,2.0]	[0.0,0.9]
RCM2M1	O	FLOAT	[0.0,1.0]	[0.0,0.5]
RSRHV	O	FLOAT	[0.0,10.0]	[2.5,4.6]
Rextrins	O	FLOAT	[0.0,10.0]	[1.6,2.3]
TOXP1N_PLUS	O	FLOAT	[0.0,1000.0]	[370.0,470,0]
B12x10_AT_10	O	FLOAT	[0.0,1000.0]	[120.0,350.0]
B12x10_AT_100	O	FLOAT	[0.0,1000.0]	[120.0,350.0]
VEARLY	O	FLOAT	[0.0,250.0]	[35.0,150.0]
CP1TOX	O	FLOAT	[500.0,3000.0]	[1816.0,2156.0]
CP1HOX	O	FLOAT	[500.0,1500.0]	[767.0,986.0]

```
%NewTest
SERIAL NUMBER    : 20220990000019
TEST MADE BY     : SR
LOCATION NAME    : CERN
Run number       : Whatever
TEST DATE        : 28/11/2001
PASSED           : Yes
PROBLEM          : NO
#-------------------------------------------------------------
#  Data part to be entered for the 5 test structures
#
%ABCDwafer
# Structure 1
# VTN08  IDN08  BVN08  VTP08  IDP08  BVP08  (6 values)
    .      .      .      .      .      .
# RSBL  RSN_PLUS  RSP_PLUS  RSP1  BETA_AT_10  BETA_AT_100  (6 values)
1.0         1.0     1.0       1.0            1.0                  1.0
# VAT  BVCEOT  BVCEST  VPJ12  IDPJ12 (5 values)
1.0         1.0     1.0          1.0       1.0
# RCM1N_PLUS  RCM1P_PLUS  RCM1P1  RCM2M1  RSRHV  Rextrins  (6 values)
1.0                 1.0               1.0        1.0          1.0     1.0
# TOXP1N_PLUS  B12x10_AT_10  B12x10_AT_100  VEARLY  CP1TOX  CP1HOX  (6 values)
1.0             1.0          1.0         1.0          1500.0       1500.0
```

Upload file – Example for wafer screening

The command line is:

java UploadTestData File SCTusername SCTpassword

Inside the file:

```
# Example of datafile
# ==================
%NewTest
SERIAL NUMBER    : 20220990304139
TEST MADE BY     : DF
LOCATION NAME    : CERN
Run number       : Here is my run number zp
TEST DATE        : 13/12/2001
PASSED           : NO
PROBLEM          : NO
#
%CHIPABCDDAT
MACHINENAME      : Machine
ONLINESOFTREV    : OnLine
OFFLINESOFTREV   : OffLine
FIRMWAREREV      : Firmware
PRFFLG           : YES
DIGFLG           : NO
DPP              : NO
# Values (128 values)
# PreampBias, ShaperBias (2)
0 0
# VDD to VddTest7F1 (18)
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
# FmaxTest1 to FmaxTest7 (7)
0 0 0 0 0 0 0
# ThrsDACSlope to Ndead (20)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
# Status to MinVswTin1 (35)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
# VswLED to Spare9 (46)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
#
%CHIPABCDCHAN
# Channel# Gain Offset Noise TrDACSlope TrDACErr status statusValid
0 040.8 -045.98 34.7 4.7 61.9 23 24
1 140.8 -145.98 35.7 4.8 67.9 24 25
2 240.8 -245.98 36.7 4.9 62.9 25 26
3 340.8 -345.98 37.7 4.3 63.9 26 27
4 440.8 -445.98 38.7 4.2 64.9 27 28
```

SCT DAQ Implementation status

Table	Table approved for addition to DB	Table created in dev DB	Web access view in dev DB	Java Upload on dev DB	Definition for test, defect and test list in dev DB[♣]	Move all in prod DB + web doc and links
TSTDAQINFO	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTDCSINFO	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTSCANINFO	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTMODIV	***Already exists in TEST_DET_IV and the test name is "DetModIV"***					
TSTHYBRESET	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBCLOCK	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBBPASS	**YES**	**YES**	**YES**	**YES**	**YES**	Not yet
TSTHYBPIPE	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBDELAY	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBRC	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBTRIM	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBNOISE	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBTW	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet
TSTHYBLTT	**YES**	**YES**	Not yet	**YES**	**YES**	Not yet

[♣] Only for the forward K5 hybrid and module defined into Test_list. It still needs to be defined for barrel module/Hybrid.