DOH Database Details

Table of contents

This document describes the processes involved in creating/uploading DOH data onto the CMS Tracker database. The general idea is to store all DOH data, including DOHs that have passed and failed tests at Kapsch and DOHs that will ultimately be used by sub-detectors other than the Tracker (Pixels, Preshower, ECAL, RPCs).

1. Introduction to CMS Tracker Database File Types

Four different types of CMS Tracker database files can be created when processing DOH data:

- **Register files**.
- **Assembly files.**
- **Action files.**
- **Transfer files.**

1.1. Register Files

Everytime a batch of DOHs is received at CERN, one registration file should be created. This should list all the DOHs in the batch. None of the other types of files for a given DOH can be uploaded to the CMS Tracker database before a registration file containing info about that given DOH has been uploaded to the database.

1.2. Assembly Files

The assembly file allows for the tracking of any objects with a CMS barcode that have been integrated into parent objects with CMS barcodes. One such file should be created for every DOH. The file should contain the DOH barcode (parent object) as well as laser and photodiode barcodes (child objects).

1.3. Action Files

The action files contain test data information. Five action files must be created for every DOH: one action file for test data generated at Kapsch and four action files for test data generated at CERN. All DOHs shipped to CERN will have Kapsch test data associated with them. It is not yet decided whether all DOHs will also be tested at CERN. If this is not the case, CERN data files should still be created for all DOHs, with blanks where the fields cannot be updated.

1.4. Transfer Files

Transfer files could be created to indicate when DOHs are given to the contact person for a given sub-detector. However, there is no provision for differentiating between sub-detectors which are based at CERN in the current database structure. CERN is listed as one center. In future, we could envisage creating other centers (e.g. CERN_TOB, CERN_TEC etc…). This would allow us to indicate that DOHs have left us and have been delivered to an assembly point somewhere at CERN. The only problem with this scheme is that the unit (e.g. control module) onto which the DOH will be mounted would have to be listed as being at the precise location that the DOH is sent to. If the responsible for that unit has declared it to be at CERN rather than CERN_TOB for example, the database will not accept an assembly file showing a module at CERN and one of its components at another location.
No transfer files are foreseen, the intended destination for each DOH will have to be known upon reception and will be declared in a field within the Kapsch data file.

2. Database File Formats

2.1. Register File Format

An example register file is given below. It shows all features of the register file:

```
<registration>
        <registrationItem>
                <commonData>
                        <date>2002-01-31T23:00:00</date>
                        <name>MOD</name>
                        <version/>
                        <type>1.1.1.2.1</type>
                        <center>LYON</center>
                <commonData>
                <objects>
                        <object id=”7700” faulty=”false”/>
                        <object id=”7701” faulty=”false”/>
                </objects>
        </registrationItem>
</registration>
```

2.2. Assembly File Format

All parent and child objects in an assembly file have to be registered in the Database. An example assembly file is shown below for one parent component (assemblyItem). Many such assemblyItems could be inserted into one assembly file. However, it turns out that there is a limit to the number of parent objects that can be inserted into an assembly file. For this reason, one assembly file should be created per parent component.

```
<assembly>
        <assemblyItem>
                <parent id=”7700”/>
                <subobject action=”add” date=”2002-02-14T16:23:58”
                        faulty=”false” id=”77002” position=”1”/>
                <subobject action=”remove” date=”2002-02-14T16:23:58”
                        faulty=”false” id=”77002” position=”1”/>
                <subobject action=”add” date=”2002-02-14T16:23:58”
                        faulty=”false” id=77001” position=”1”/>
                <subobject action=”add” date=”2002-02-14T16:23:58”
                        faulty=”false” id=”77002” position=”2”/>
                <subobject action=”add” date=”2002-02-14T16:23:58”
                        faulty=”false” id=”77003” position=”3”/>
        </assemblyItem>
<assembly>
```

2.3. Action XML File Format

The xml file format for data files stored on the CMS database is the following:

- Unit node
    - Object node (object id)
    - Composite node (at least one)
        - Action description (mandatory)
        - Composite node
        - Action node
            - Action description
            - Input node (any number)
            - Result node (any number)

A brief description of these objects is given below:

- A unit node is made of an object node and at least one composite node
- an object node has one argument, the object id
- a composite node has three children nodes : an action description (mandatory) and at least one action or composite node.
- an action_description node has up to 6 arguments (e.g. name, version, object_name, object_type and input_id)
- an action node can have two children node types : input node and result node. The number of occurences of these two nodes is totally free.
- An input node is a list of two arguments: the name of the input parameter and it's value
- An result node is a list of two arguments: the name of the result and it's value

3. Creation of Database Files

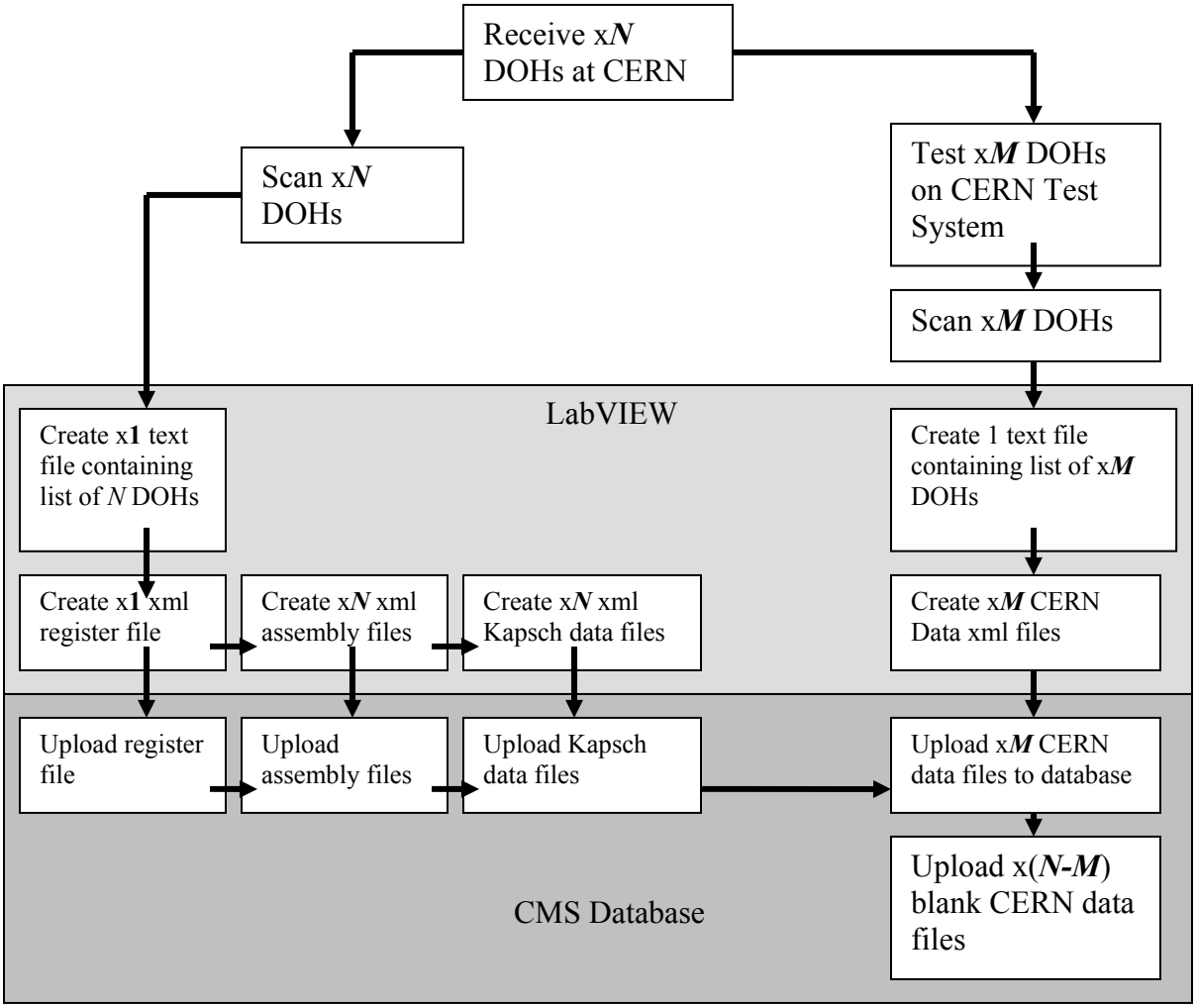

Figure 1: Creation and uploading of DOH database files.

The creation of all database files will be done using LabVIEW.
Upon reception of the DOHs at CERN, the proposal is to scan every single DOH with a cordless scanner. Following this step, a text file containing all the DOH barcodes is produced. This text file is then used:

- to create one .xml register file containing all the DOH barcodes
- to create one .xml assembly file containing all DOH, laser and PD barcodes
- to create one .xml file per DOH containing Kapsch test data for every DOH scanned.

Once a given number (to be defined) of DOHs have been tested on the CERN test system, they should be scanned with a cordless scanner. A text file containing all DOH barcodes is again produced (different to the text file created for the DOHs tested at Kapsch). This text file is used

by a LabVIEW program to create four .xml files containing CERN test data for every DOH scanned.

4. Uploading XML Files to the Database

The five xml files created per DOH add up to ~87KB. The PC used for file uploads is essentially un-useable for other purposes during the upload periods which will probably be lengthy due to the large numbers of DOHs. It is therefore recommended to carry out the file uploads with a PC which is not being used regularly (e.g. lab PC).
Register files are uploaded by selecting the 'Open' menu item from the CMS Database browser and selecting the file to be uploaded.
Assembly files should be stored in a result ***C:\Users\Etam\TrackerDB\upload\results***. They cannot be uploaded in the same way as action files but a LabVIEW vi, InsertXMLFilesAutomatically.vi, in the trackerDBqueries.llb library will do the same job.
All action xml files created by LaBVIEW will be stored in the following directory: ***C:\Users\Etam\TrackerDB\upload\results.***
This directory should be set as the default upload directory in the CMS Database browser
Action files are updated by selecting the ***Update local files to DB*** menu item in the ***DB Update*** menu. The action files are parsed and updated in the DB.
The log file DBupdate.log is updated and faulty transferred files are marked by an ERROR flag in the log file, while successful transferred files are marked with an OK flag, renamed with a `.indb` suffix and stored in the following directory on the local machine: ***C:\Users\Etam\TrackerDB\upload\indb.***

5. Details of DOH Action Files

The DOH action files will be written with the following structure:

- PRODUCTION (Composite node)
  - KAPSCHTESTDATA (Action node)
  - CERNTESTSUMMARY (Action node)
  - CERNTESTTX (Action node)
  - CERNTESTRX (Action node)
  - CERNTESTPOWERSUPPLY (Action node)

The files will be written in two steps. The first step takes place upon reception of the DOHs and consists of writing the KAPSCHTEST_DATA action node. The second step takes place after having tested all DOHs and consists of writing the 4 CERNTEST action nodes listed above.

5.1. Kapsch Data File Structure

- DOHPRODUCTION (action description for composite node)
    - KAPSCHTESTDATA (action description for action node)

	Field Name	Field type	Units	Description
1	TDATE	date		Date at which DOH was tested at Kapsch
2	OPERATOR	varchar(32)		Initials of person responsible for uploading files to the database. Default is EN
3	TOOL_ID	int		Identification number for the test bench used at Kaspch, value is 903
4	BATCH	int		Batch number for the given DOH
5	PIGTAIL_LENGTH	float	cm	Pigtail length
6	DELIVERY_DATE	date		Date DOH was delivered to CERN
7	XML_DATE	date		Creation date of XML file
8	KAPSCH_TEST_STATUS	int		Summary of Kapsch test. FAIL = 0, PASS = 1
9	REPAIR	varchar(32)		Description of repair process
10	DESTINATION	varchar(32)		Destination of DOH
11	KAPSCH_TEST_NUMBER	int		Kapsch test reference number assigned during testing process
12	CLOCK_RX40_TEST	int		Result for Clock receiver tests. FAIL = 0, PASS = 1
13	DATA_RX40_TEST	int		Result for Data receiver tests. FAIL = 0, PASS = 1
14	CLOCK_LLD_TEST	int		Result for Clock transmitter tests. FAIL = 0, PASS = 1
15	DATA_LLD_TEST	int		Result for Data transmitter tests. FAIL = 0, PASS = 1
16	RESET_TEST	int		Result for reset test. FAIL = 0, PASS = 1
17	I2C_TEST	int		Result for I2C test
18	CLOCK_AMP_GAIN0	float	dBm	Clock laser OMA for LLD gain set to 0
19	CLOCK_AMP_GAIN1	float	dBm	Clock laser OMA for LLD gain set to 1
20	CLOCK_AMP_GAIN2	float	dBm	Clock laser OMA for LLD gain set to 2
21	CLOCK_AMP_GAIN3	float	dBm	Clock laser OMA for LLD gain set to 3. This is also the default OMA.
22	DATA_AMP_GAIN0	float	dBm	Data laser OMA for LLD gain set to 0
23	DATA_AMP_GAIN1	float	dBm	Data laser OMA for LLD gain set to 1
24	DATA_AMP_GAIN2	float	dBm	Data laser OMA for LLD gain set to 2
25	DATA_AMP_GAIN3	float	dBm	Data laser OMA for LLD gain set to 3. This is also the default OMA.
26	CLOCK_MEAS_LASERBIAS	varchar(4000)	I2C	Clock LLD bias. Plot 27 (y axis) vs 26 (x axis).
27	CLOCK_MEAS_LASEROUTPUT	varchar(4000)	μW	Clock laser output for different I2C values. Plot 27 (y axis) vs 26 (x axis).
28	DATA_MEAS_LASERBIAS	varchar(4000)	I2C	Data LLD bias. Plot 29 (y axis) vs 28 (x axis).
29	DATA_MEAS_LASEROUTPUT	varchar(4000)	μW	Data laser output for different I2C values. Plot 29 (y axis) vs 28 (x axis).
30	PD_TIME	varchar(4000)	s	Time axis useful for plotting RX40 electrical outputs, 31 and 32
31	CLOCK_PD_ELEC_OUTPUT	varchar(4000)	V	Clock RX40 electrical output. Plot 31 (y axis) vs 30 (x-axis)
32	DATA_PD_ELEC_OUTPUT	varchar(4000)	V	Data RX40 electrical output. Plot 32 (y axis) vs 30 (x-axis)
33	KAPSCHTESTDATA_VAL	varchar(32)		
34	STATUS	varchar(32)		
35	TCOMMENT	varchar(32)		

All numbered elements in the list above are result nodes. Note that the fields in blue are mandatory. TDATE here is the test date (i.e. the date the DOH was tested at Kapsch). There are 3 tool_id, one for the Kapsch Main test setup (located at Kapsch), one for the Kapsch Clone test setup (located at CERN) and finally one for the CERN test setup (located at CERN).
Fields 1, 5, 8, 11-32 are read automatically from the Kapsch data file. Field 7 is returned by the LabVIEW vi that creates the data xml file. All other fields (2, 3, 4, 6, 9, 10, 33, 34, 35) have to be edited in the LabVIEW vi before it creates the xml file.
The repair control within the LabVIEW vi that creates the xml file will have a list of all possible repairs. It will also be complemented by the destination control which will have standard destinations (CERN_TOB, CERN_TEC etc…) as well as repair destinations (CERN_REPAIR, KAPSCH_REPAIR).

5.2. CERN Data File Structure

The CERN test data file is broken down into four sub-actions for ease of navigation through the different types of fields for eventual users of the database:

- DOHPRODUCTION
    - CERNTESTSUMMARY
    - CERNTESTTX
    - CERNTESTRX
    - CERNTESTPOWERSUPPLY

The details of the file structure including all fields are the following:

- DOHPRODUCTION

- CERNTESTSUMMARY

	Field Name	Field type	Units	Description
1	TDATE	date		Date at which DOH was tested at CERN
2	OPERATOR	varchar(32)		Initials of person responsible for uploading files to the database. Default is EN
3	TOOL_ID	int		Identification number for the test bench used at CERN, value is 905
4	BATCH	int		Batch number for the given DOH
5	PIGTAIL_LENGTH	float	cm	Pigtail length
6	DELIVERY_DATE	date		Date DOH was delivered to CERN
7	XML_DATE	date		Creation date of XML file
8	CERN_TEST_STATUS	int		Summary of CERN test. FAIL = 0, PASS = 1
9	CERN_TEST_NUMBER	int		CERN test reference number assigned during testing process
10	CLOCK_OMA_ALP	int		Result for Clock laser default output tests. FAIL = 0, PASS = 1
11	DATA_OMA_ALP	int		Result for Data laser default output tests. FAIL = 0, PASS = 1
12	CLOCK_LI	int		Result for Clock L-I characteristics. FAIL = 0, PASS = 1
13	DATA_LI	int		Result for Data L-I characteristics. FAIL = 0, PASS = 1
14	CLOCK_VOLTAGE_SWING	int		Result for Clock RX40 output. FAIL = 0, PASS = 1
15	DATA_VOLTAGE_SWING	int		Result for Data RX40 output. FAIL = 0, PASS = 1
16	CLOCK_SENSITIVITY	int		Result for Clock photodiode sensitivity. FAIL = 0, PASS = 1
17	DATA_SENSITIVITY	int		Result for Data photodiode sensitivity. FAIL = 0, PASS = 1
18	CLOCK_SATURATION	int		Result for Clock receiver saturation. FAIL = 0, PASS = 1
19	DATA_SATURATION	int		Result for Data receiver saturation. FAIL = 0, PASS = 1
20	DOH_RESET	int		Result for DOH reset test. FAIL = 0, PASS = 1
21	I2C_TEST	int		Result for DOH I2C test. FAIL = 0, PASS = 1
22	POWER_SUPPLY	int		Result for Power Supply tests. FAIL = 0, PASS = 1
23	CERNTESTSUMMARY_VAL	varchar(32)		
24	STATUS	varchar(32)		
25	TCOMMENT	varchar(32)		

- CERNTESTTX

	Field Name	Field type	Units	Description
1	TDATE	date		Date at which DOH was tested at CERN
2	OPERATOR	varchar(32)		Initials of person responsible for uploading files to the database. Default is EN
3	TOOL_ID	int		Identification number for the test bench used at CERN, value is 905
4	CERN_TEST_NUMBER	int		CERN test reference number assigned during testing process
5	CLOCK_TX_OMA_DEFAULT	float	dBm	Default Clock laser OMA
6	DATA_TX_OMA_DEFAULT	float	dBm	Default Data laser OMA
7	CLOCK_TX_ALP_DEFAULT	float	dBm	Default Clock laser ALP
8	DATA_TX_ALP_DEFAULT	float	dBm	Default Data laser ALP
9	LASER_TX_OUTPUT_TIME	varchar(4000)	s	Time axis useful for plotting laser outputs, 10 and 11
10	CLOCK_TX_LASER_DEFAULT_WFM	varchar(4000)	µW	Default Clock laser output, plot 10 (y-axis) vs 11 (x-axis)
11	DATA_TX_LASER_DEFAULT_WFM	varchar(4000)	µW	Default Data laser output, plot 10 (y-axis) vs 11 (x-axis)
12	CLOCK_TX_AMP_GAIN0	float	µW	Clock laser OMA for LLD gain set to 0, in units of dBm
13	CLOCK_TX_AMP_GAIN1	float	µW	Clock laser OMA for LLD gain set to 1, in units of dBm
14	CLOCK_TX_AMP_GAIN2	float	µW	Clock laser OMA for LLD gain set to 2, in units of dBm
15	CLOCK_TX_AMP_GAIN3	float	µW	Clock laser OMA for LLD gain set to 3, in units of dBm. Default OMA.
16	DATA_TX_AMP_GAIN0	float	µW	Data laser OMA for LLD gain set to 0, in units of dBm
17	DATA_TX_AMP_GAIN1	float	µW	Data laser OMA for LLD gain set to 1, in units of dBm
18	DATA_TX_AMP_GAIN2	float	µW	Data laser OMA for LLD gain set to 2, in units of dBm
19	DATA_TX_AMP_GAIN3	float	µW	Data laser OMA for LLD gain set to 3, in units of dBm. Default OMA.
20	CLOCK_TX_LI_LASERBIAS	varchar(4000)	I2C	Clock LLD bias in I2C units. Plot of 21 (y axis) vs 20 (x axis).
21	CLOCK_TX_LI_LASEROUTPUT	varchar(4000)	µW	Clock laser ALP for different I2C values. Plot 21 (y axis) vs 20 (x axis).
22	DATA_TX_LI_LASERBIAS	varchar(4000)	I2C	Data LLD bias in I2C units. Plot 21 (y axis) vs 20 (x axis).
23	DATA_TX_LI_LASEROUTPUT	varchar(4000)	µW	Data laser ALP for different I2C values. Plot 21 (y axis) vs 20 (x axis).
24	CLOCK_TX_OMA_LASERBIAS	varchar(4000)	I2C	Clock LLD bias in I2C units. Use to plot 25, 26, 27, 28
25	CLOCK_TX_OMA_LASEROUTPUT	varchar(4000)	µW	Clock laser OMA for different I2C bias values, default gain=3. Plot 25 vs 24
26	CLOCK_TX_OMA_G0_LASEROUTPUT	varchar(4000)	µW	Clock laser OMA for different I2C bias values, gain=0. Plot 26 (y-axis) vs 24 (x-axis)
27	CLOCK_TX_OMA_G1_LASEROUTPUT	Varchar(4000)	µW	Clock laser OMA for different I2C bias values, gain=1. Plot 27 (y-axis) vs 24 (x-axis)
28	CLOCK_TX_OMA_G2_LASEROUTPUT	Varchar(4000)	µW	Clock laser OMA for different I2C bias values, gain=2. Plot 28 (y-axis) vs 24 (x-axis)
29	DATA_TX_OMA_LASERBIAS	varchar(4000)	I2C	Data LLD bias in I2C units. Use to plot 30, 31, 32, 33
30	DATA_TX_OMA_LASEROUTPUT	varchar(4000)	µW	Data laser OMA for different I2C values. Plot 30 (y-axis) vs 29 (x-axis).
31	DATA_TX_OMA_G0_LASEROUTPUT	varchar(4000)	µW	Data laser OMA for different I2C bias values, gain=0. Plot 31 (y-axis) vs 29 (x-axis)
32	DATA_TX_OMA_G1_LASEROUTPUT	Varchar(4000)	µW	Data laser OMA for different I2C bias values, gain=1. Plot 32 (y-axis) vs 29 (x-axis)
33	DATA_TX_OMA_G2_LASEROUTPUT	Varchar(4000)	µW	Data laser OMA for different I2C bias values, gain=2. Plot 33 (y-axis) vs 29 (x-axis)
34	CERNTESTTX_VAL	varchar(32)		
35	STATUS	varchar(32)		
36	TCOMMENT	varchar(32)		

- CERNTESTRX

	Field Name	Field type	Units	Description
1	TDATE	date		Date at which DOH was tested at CERN
2	OPERATOR	varchar(32)		Initials of person responsible for uploading files to the database. Default is EN
3	TOOL_ID	int		Identification number for the test bench used at CERN, value is 905
4	CERN_TEST_NUMBER	int		CERN test reference number assigned during testing process
5	CLOCK_RX_OUT_DIFF_VALUE	float	V	Clock RX40 differential output amplitude, in units of V
6	DATA_RX_OUT_DIFF_VALUE	float	V	Data RX40 differential output amplitude, in units of V
7	CLOCK_RX_OUT_POS_VALUE	float	V	Clock RX40 positive output amplitude, in units of V
8	DATA_RX_OUT_POS_VALUE	float	V	Data RX40 positive output amplitude, in units of V
9	CLOCK_RX_OUT_NEG_VALUE	float	V	Clock RX40 negative output amplitude, in units of V
10	DATA_RX_OUT_NEG_VALUE	float	V	Data RX40 negative output amplitude, in units of V
11	CLOCK_RX_OUT_TIME	varchar(4000)	s	Time axis useful for plotting RX40 Clock output waveforms, 12, 13 and 14
12	CLOCK_RX_OUT_DIFF_WFM	varchar(4000)	V	RX40 Clock differential output waveform, plot 12(y-axis) vs 11(x-axis)
13	CLOCK_RX_OUT_POS_WFM	varchar(4000)	V	RX40 Clock positive output waveform, plot 13(y-axis) vs 11(x-axis)
14	CLOCK_RX_OUT_NEG_WFM	varchar(4000)	V	RX40 Clock negative output waveform, plot 14(y-axis) vs 11(x-axis)
15	DATA_RX_OUT_TIME	varchar(4000)	s	Time axis useful for plotting RX40 Data output waveforms, 16, 17 and 18
16	DATA_RX_OUT_DIFF_WFM	varchar(4000)	V	RX40 Data differential output waveform, plot 16(y-axis) vs 15(x-axis)
17	DATA_RX_OUT_POS_WFM	varchar(4000)	V	RX40 Data positive output waveform, plot 17(y-axis) vs 15(x-axis)
18	DATA_RX_OUT_NEG_WFM	varchar(4000)	V	RX40 Data negative output waveform, plot 18(y-axis) vs 15(x-axis)
19	CLOCK_RX_SENS_VALUE	float	dBm	Clock receiver sensitivity, in units of dBm
20	DATA_RX_SENS_VALUE	float	dBm	Data receiver sensitivity, in units of dBm
21	CLOCK_RX_SENS_ATTN	float	dB	VOA attenuation used in test of Clock receiver sensitivity, in units of dB
22	DATA_RX_SENS_ATTN	float	dB	VOA attenuation used in test of Data receiver sensivitiy, in units of dB
23	CLOCK_RX_SENS_REF_INPUT	float	μW	Reference input used in test of Clock receiver sensitivity, in units of uW
24	DATA_RX_SENS_REF_INPUT	float	μW	Reference input used in test of Data receiver sensitivity, in units of uW
25	RX_SENS_REFDOH_TIME	varchar(4000)	s	Time axis useful to plot reference signal waveforms used in sensitivity tests, 26 and 27
26	CLOCK_RX_SENS_REFDOH_WFM	varchar(4000)	μW	Reference signal waveform used as input for clock sensitivity test, plot 26 (y-axis) vs 25
27	DATA_RX_SENS_REFDOH_WFM	varchar(4000)	μW	Reference signal waveform used as input for data sensitivity test, plot 27 (y-axis) vs 25
28	CLOCK_RX_SAT_DIFF_VALUE	float	V	RX40 Clock differential output amplitude for maximum specified input
29	DATA_RX_SAT_DIFF_VALUE	float	V	RX40 Data differential output amplitude for maximum specified input
30	CLOCK_RX_SAT_REF_IN	float	dBm	Reference signal amplitude used during Clock saturation test
31	DATA_RX_SAT_REF_IN	float	dBm	Reference signal amplitude used during Data saturation test
32	RX_SAT_DIFF_TIME	varchar(4000)	s	Time axis useful to plot RX40 differential output waveforms for saturation input, 33 and 34
33	CLOCK_RX_SAT_DIFF_WFM	varchar(4000)	V	RX40 Clock differential output waveform for saturation input signal, plot 33 vs 32
34	DATA_RX_SAT_DIFF_WFM	varchar(4000)	V	RX40 Data differential output waveform for saturation input signal, plot 34 vs 32
35	RX_SAT_REFDOH_TIME	varchar(4000)	s	Time axis useful to plot reference signal used in saturation tests,
36	CLOCK_RX_SAT_REFDOH_WFM	varchar(4000)	μW	Reference signal waveform used in Clock saturation test, plot 36 vs 35
37	DATA_RX_SAT_REFDOH_WFM	varchar(4000)	μW	Reference signal waveform used in Data saturation test, plot 37 vs 35
38	CERNTESTRX_VAL	varchar(32)		
39	STATUS	varchar(32)		
40	TCOMMENT	varchar(32)		

- CERNTESTPOWERSUPPLY

	Field Name	Field type	Units	Description
1	TDATE	date		Date at which DOH was tested at CERN
2	OPERATOR	varchar(32)		Initials of person responsible for uploading files to the database. Default is EN
3	TOOL_ID	int		Identification number for the test bench used at CERN, value is 905
4	CERN_TEST_NUMBER	int		CERN test reference number assigned during testing process
5	POWER_CONSUMPTION_DEFAULT	float	W	DOH default power consumption
6	POWER_CONSUMPTION_MAX	float	W	DOH maximum power consumption, with both active LLD channels set to max. bias of 126
7	POWER_CONSUMPTION_MIN	float	W	DOH minimum power consumption, with both LLD channels set to minimum bias of 126
8	PS_MAX_TIME	varchar(4000)	s	Time axis used to plot waveforms obtained with max DOH power supply 9, 10, 11 and 12
9	CLOCK_RX_PS_MAX_WFM	varchar(4000)	V	RX40 Clock output waveform for DOH power supply set to max of 2.7 V, plot 9 vs 8
10	DATA_RX_PS_MAX_WFM	varchar(4000)	V	RX40 Data output waveform for DOH power supply set to max of 2.7 V, plot 10 vs 8
11	CLOCK_TX_PS_MAX_WFM	varchar(4000)	μW	Clock laser output waveform for DOH power supply set to max of 2.7 V, plot 11 vs 8
12	DATA_TX_PS_MAX_WFM	varchar(4000)	μW	Data laser output waveform for DOH power supply set to max of 2.7 V, plot 12 vs 8
13	PS_MIN_TIME		s	Time axis used to plot waveforms obtained with min DOH power supply 14, 15, 16 and 17
14	CLOCK_RX_PS_MIN_WFM	varchar(4000)	V	RX40 Clock output waveform for DOH power supply set to min of 2.25 V, plot 14 vs 13
15	DATA_RX_PS_MIN_WFM	varchar(4000)	V	RX40 Data output waveform for DOH power supply set to min of 2.25 V, plot 15 vs 13
16	CLOCK_TX_PS_MIN_WFM	varchar(4000)	μW	Clock laser output waveform for DOH power supply set to min of 2.25 V, plot 16 vs 13
17	DATA_TX_PS_MIN_WFM	varchar(4000)	μW	Data laser output waveform for DOH power supply set to min of 2.25 V, plot 17 vs 13
18	CERNTESTPOWERSUPPLY_VAL	varchar(32)		
19	STATUS	varchar(32)		
20	TCOMMENT	varchar(32)		

6. Input and Tool IDs

The following Tool_IDs have been defined for the various DOH testers:

TOOL: DOH_KAPSCH_main
Center: CERN
OPERATION_TYPE: DOH Production Test
DESCRIPTION: Company setup
TOOL_ID : 903

TOOL: DOH_KAPSCH_clone
Center: CERN
OPERATION_TYPE: DOH Production Test
DESCRIPTION: CERN copy of Company setup
TOOL_ID : 904

TOOL: DOH_CERN
Center: CERN
OPERATION_TYPE: DOH Verification Test
DESCRIPTION: CERN extensive test setup
TOOL_ID : 905

The following definitions have been defined for objects and subobjects to be included in the assembly files:

OBJECT: DOH
TYPE: 1
VERSION: <empty>
SUB_OBJECT: LASTRANS
SUB_OBJECT_TYPE: 1
SUB_OBJECT_VERSION: <empty>
SUB_OBJECT_NUMBER: 2
DESCRIPTION: <empty>

OBJECT: DOH
TYPE: 1
VERSION: <empty>
SUB_OBJECT: PD
SUB_OBJECT_TYPE: 1
SUB_OBJECT_VERSION: <empty>
SUB_OBJECT_NUMBER: 2
DESCRIPTION: <empty>

Three iterations were needed on the test database to get things right, hence the three versions of input_ids. Everytime a table is changed, the input_ids for the composite node and all action nodes have to be changed.
Test database input_Ids, Version 1:

- CERNTESTPOWERSUPPLY : 1454
- CERNTESTRX : 1455
- CERNTESTSUMMARY : 1456
- CERNTESTTX : 1457

- KAPSCHTESTDATA : 1458
- DOHPRODUCTION : 1459

Test database input_Ids, Version 2:
- CERNTESTPOWERSUPPLY_2_DOH_ : 1517
- CERNTESTRX_2_DOH_ : 1518
- CERNTESTSUMMARY_2_DOH_ : 1519
- CERNTESTTX_2_DOH_ : 1520
- KAPSCHTESTDATA_2_DOH_ : 1521
- DOHPRODUCTION_2_DOH_ : 1522

Test database input_Ids, Version 3:
- CERNTESTPOWERSUPPLY_3_DOH_ : 1528
- CERNTESTRX_3_DOH_ : 1529
- CERNTESTSUMMARY_3_DOH_ : 1533
- CERNTESTTX_3_DOH_ : 1531
- KAPSCHTESTDATA_3_DOH_ : 1534
- DOHPRODUCTION_3_DOH_ : 1535

As well as the three input_Ids for the test database, the following input_Ids were issued for the production database:
Version 1:
- CERNTESTPOWERSUPPLY_1_DOH_ : 1098
- CERNTESTRX_1_DOH_ : 1099
- CERNTESTSUMMARY_1_DOH_ : 1100
- CERNTESTTX_1_DOH_ : 1101
- KAPSCHTESTDATA_1_DOH_ : 1102
- DOHPRODUCTION_1_DOH_ : 1103

7. Dealing with faulty DOHs

If a DOH is faulty before its details are loaded into the database, its faulty state should be declared in the registration file by setting ***faulty='true'***. If a DOH is faulty after having had its details loaded into the database or if its faulty status was wrong in the registration file, it can be changed by applying the following steps:
- from the database front panel, select Worksation > special action
- type in the DOH barcode and press 'enter'
- the date will show up automatically
- type the operator name e.g. EN
- type a description of why the status is being changed, e.g. 'Faulty DOH, Bad RX Channel' in the Results section
- Select the appropriate quality flag.
- Set status to 'reference'
- Click 'Update'

Note that if a DOH is faulty, the test flags must be set to -1 in all of the 5 action tables. E.g. cerntesttx_val=':-1:'. Actually, this doesn't have to be so, since changing just one of the tables will automatically set DOHPRODUCTION_ to faulty for that object but for the DOH, it was chosen to set all tables to -1.

The status of all action files must always be reference. If an action file is loaded twice, the status of the first action file loaded is changed automatically by the database to 'valid', whilst the second file loaded becomes the 'reference' file.
If a DOH was initially declared faulty in the database and is subsequently repaired, loading new action files with repaired data is not sufficient to change the DOH from faulty to valid. An action REPAIR is needed to change the state of the DOH in the database (Workstation > repair object)

8. Using the Production Database

This section gives a brief guide to using the production database to search for information on the DOHs. One of the following passwords is required to access the production database:

- read-only : RdBP!$p
- update: UdBP!$p

8.1. Viewing tables

To look at all the information stored in the tables, select *Tables > View Tables*, from the database front panel. More details on the following tables can be obtained by double-clicking on the table:

- CERNTESTPOWERSUPPLY_1_DOH_
- CERNTESTRX_1_DOH_
- CERNTESTSUMMARY_1_DOH_
- CERNTESTTX_1_DOH_
- DOHPRODUCTION_1_DOH_
- KAPSCHTESTDATA_1_DOH_

8.2. Viewing Statistics

Statistics on any of the field types that are either 'int' or 'float', can be looked at by selecting *Quality Control > Statistics* from the database front panel.

8.3. Viewing Vectors

Vectors in the database are described by the field type 'varchar(4000)' and can be looked at by selecting *Quality Control > Vectors* from the database front panel. When trying to plot an *x action* vs a *y action*, follow the rules listed in the 'description' column of the tables in section 5.

8.4. Free SQL queries

Free SQL queries are useful for performing more detailed searches through the database. E.g. *select batch,count(*) from kapschtestdata_1_doh_ where status='reference' group by batch* will return a table summarising how many DOHs are in each batch.
A list of sql queries is provided in the following bookmarks file: *bookmarks_sql_queries.txt* under the TrackerDB directory.

8.5. Inventory

The inventory section is useful to quickly obtain a summary of all DOHs in the database and their assembly and faulty status.

8.6. State of DOHs

To see a list of the number of DOHs separated between those that are ‘running’ (i.e. not all action files with a ‘reference’ status have been loaded into the database for that object), those that are faulty (i.e. a negative flag :-1: exists for one of the 5 action tables) and those that are ready do *Production status* > *report*, select ‘*tests*’ and ‘*DOHPRODUCTION*’, set appropriate dates to include all files, select ‘*Object count*’ (on the left of the screen), select ‘*group by object*’.