ATLAS Internal Note

1st of February 2007

Tile Online Status Browser Application

***C. Solans*[1]**, *J. Abdallah*[1], *V. Castillo*[1], *C. Cuenca*[1], *A. Ferrer*[1], *E. Fullana*[1], *V.González*[2], *E. Higón*[1], *J. Poveda*[1], *A. Ruiz-Martínez*[1], *B. Salvachúa*[1], *E.Sanchís*[2], *O. Solov'yanov*[3], *J.Torres*[2], *A. Valero*[1], *J. Valls*[1]

[1] Departamento de Física Atómica, Molecular y Nuclear and IFIC, Aptdo. 22085, Valencia, España
[2] Departamento de Electrónica, Universidad de Valencia, España

[3] IHEP. Russian Fed. State Res.Cent. Protvino, Russia

ATL-TILECAL-INT-2007-002
03 March 2007

Abstract

This document describes the Tile Online Status browser application. A web interface where the information relative to the different commissioning teams of the ATLAS Tile Hadronic Calorimeter is stored and retrieved. The information contains the status of the super-drawer and the comments relative to their performance. This is displayed in wedged barrels of multiple layers or in a coloured table. This document is meant to be a reference for future Tile online applications.

Index

1 Introduction

The ATLAS Hadronic Tile Calorimeter (TileCal) detector is divided into four barrels. Each barrel is subdivided into 64 modules. The front-end and digitizing electronics are situated in the back-beam region inside each module in movable griders, the so-called super-drawers. Currently the TileCal is undergoing the commissioning phase of these super-drawers.

This phase of the TileCal detector involves many working teams together. The upstream of information coming from each working group must be easily accesible from to everybody in order to coordinate all commissioning tasks. We summarize the information gruped by super-drawer for all the super-drawers in one single place. This information is the status and the comments on the performance of the super-drawer relative to each of the commissioning teams.

Furthermore, the need for a tool that will allow quick display of the information relative to the TileCal detector in a graphical way through the web is motivation enough to develop a tool for it. This tool is implememented in the Tile Online Status browser and it will be described.

The Tile Online Status browser application is a web interface based on Hypertext Preprocessor (PHP) scripting language to display the information, Graphics Library (GD) to draw the TileCal barrel status pictures and a MySQL relational database server (MySQL) to store the information.

1.1 Requirements

The requirements for the Tile Online Status browser application are the following.

- Display information through a coloured list and a wedged barrel.
- Allow users to change the status of the pieces of information
- Simple clickable interface
- Status information and small comments for four online groups: LV, HV, Team4 (online data taking), Team5 (offline QA of the data).
- No history. Keep date from last modification.
- Simple color schema: White, green, yellow, orange, red.
- Defined states: Undefined, good, investigating, minor damages, major damages.
- Allow small comments for each group and module
- Web server with PHP version grater than 4.X with GD and MySQL extensions installed.

2 Back–end database

The information stored in the database is transparent for the user. However, the structure of this information complies with the project requirements to ease the design. The information is grouped into one single table where the super-drawer name is the index. As no history is requiered, the super-

drawer index key is also unique. Meaning there cannot be two entries in the database with the same super-drawer.

According to the requirements, the online status for each of the commissioning groups is defined as an enumeration type and the comments for each group is defined as a string type.

The database unique table schema can be resumed in the following table where the table fieldnames, types and default values are shown. These headers correspond to the ATLAS Trigger and DAQ system (TDAQ) naming convention:

Name	Type	Range	Init-value
Drawer	String		
Lv	Enumeration	undefined, good, investigating, minor_damages, major_damages	Undefined
lvcomments	String		
Hv	Enumeration	undefined, good, investigating, minor_damages, major_damages	Undefined
hvcomments	String		
Team4	Enumeration	undefined, good, investigating, minor_damages, major_damages	Undefined
Team4comments	String		
Team5	Enumeration	undefined, good, investigating, minor_damages, major_damages	Undefined
Team5comments	String		

Table 1: Database table schema.

The status states for each of the table elements are related to each colour like the following.

Status	Colour
Undefined	White
Good	Green
Investigating	Yellow
Minor damages	Orange
Major damages	Red

Table 2: Status elements table schema.

MySQL is a relational database server. There is no need to develop any other kind of database storage as MySQL is license free and easy to setup and deploy. Also, PHP provides a native extension for MySQL in such a way there is no special development to do.

Consequently, the information from the Tile Online Status browser application is stored in a MySQL database. Accordingly, the schema types have to match MySQL types. The following table describes the simple correspondencies.

Schema	MySQL
String	Varchar
Enumeration	Enum

Table 3. Schema – MySQL type correspondencies.

2.1 SQL commands

MySQL as any other relational database works with Simple Query Language (SQL) queries. The user submits a query to the MySQL server engine and the engine returns the according answer to the user.

Nevertheless, the role of the user is very important in the MySQL database administration. By default, there is a root user with all privileges granted. The advice is to create a user with access privileges to a given database and use it to do all the insertion, selection and update of the data.

The fact that the user has control over the server where the database is stored will be asumed. As CERN does not officially support MySQL installations, the MySQL installation has to be done in an end user machine. The installation of a MySQL server on a Scientific Linux CERN (SLC) distribution is setup as the root user through the commands.

```
apt-get install mysql-server

chkconfig mysqld on

service mysqld start
```

In the following, the SQL commands needed to create the database and the table, create the user and grant access to it from the root user will be described. They are all executed from the MySQL shell. Also, the SQL commands requiered by the front-end application to insert the data, retrieve and update it will be described. The commands will be shown for a set of predefined parameters. Table 4 shows the list of parameters that will be used.

Parameter	Value
database	tileonlinestatus
user	tilecal
password	drawerstatus

Table 4: SQL command parameters

The same installation can be done in any other server replacing the parameters with other values.

2.1.1 Database creation

From the MySQL shell running as root in the database server machine, the command to create the Tile Online Status MySQL database from scratch is:

```
CREATE DATABASE tileonlinestatus;
```

The query will return a zero statement with the time it took to execute the query.

2.1.2 User creation

Once the MySQL database has been created, the user and the grant access for insertion, selection and update have to be created. From the MySQL shell running as root in the database server machine, the command to create a MySQL user and grant access to it for a given database is:

```
GRANT SELECT,INSERT,UPDATE,DELETE
ON tiledrawerstatus.*
TO tilecal
IDENTIFIED BY 'drawerstatus';
```

The query will return a zero statement with the time it took to execute the query.

2.1.3 Table creation

The following SQL commands will be executed from the front end web application as the granted user. The command to create the table for the Tile Online Status browser application is:

```
CREATE TABLE IF NOT EXISTS tileonlinestatus (
`drawer` VARCHAR(5) NOT NULL,

`team5` ENUM('undefined','good','investigating','minor_damages','major_damages') NOT
NULL DEFAULT 'undefined' ,

`team5comments` VARCHAR(255) NOT NULL,

`team4` ENUM('undefined','good','investigating','minor_damages','major_damages') NOT
NULL DEFAULT 'undefined' ,

`team4comments` VARCHAR(255) NOT NULL,

`lv` ENUM('undefined','good','investigating','minor_damages','major_damages') NOT NULL
DEFAULT 'undefined' ,

`lvcomments` VARCHAR(255) NOT NULL,

`hv` ENUM('undefined','good','investigating','minor_damages','major_damages') NOT NULL
DEFAULT 'undefined' ,

`hvcomments` VARCHAR(255) NOT NULL,

PRIMARY KEY (`drawer`)

);
```

2.1.4 Data insertion

As the default state for each group is undefined and the default value for each state field in the database is undefined, the insertion of the defatult value can be omitted in the SQL insertion command. The same can be extended for the comment field for each group. Finally, the command to insert a new entry in the table for the Tile Online Status browser application is:

```
INSERT INTO tileonlinestatus (`drawer`) VALUES ('%drawer_name%');
```

Where the %drawer_name% variable has to be replaced by the correct super-drawer name. This is the five character composition of the name: AAA00. Where AAA is EBA, EBC, LBA, LBC and 00 is a two digit number from 1 to 64.

2.1.5 Data selection

One of the most important SQL commands to be performed from the Tile Online Status browser application is the select command. The following is the SQL syntax to retrieve information from the database.

```
SELECT * FROM tileonlinestatus WHERE drawer LIKE '%drawer_name%';
```

Where the %drawer_name% variable has to be replaced by the correct super-drawer name. AAA00 as exlained in [2.1.4].

This SQL command will retrieve a resultset with one single entry from the database. However, a similar SQL command can be executed to retrieve the entries for a whole barrel. Thus:

```
SELECT * FROM tileonlinestatus WHERE drawer LIKE 'AAA%';
```

Where AAA has to be replaced by the barrel name (LBA, LBC, EBA, EBC) and the last % character kept.

2.1.6 Data update

The SQL data update command requieres some special treatment. The general strucutre for the update SQL command is:

```
UPDATE tileonlinestatus SET %fields% WHERE drawer='%drawer_name%';
```

Where the %fields% variable is comma separated list constructed like the following

```
%fieldname% = '%fieldvalue%'
```

And the %drawer_name% variable has to be replaced by the correct super-drawer name. AAA00 as exlained in [2.1.4].

The following example will update the status of the team4 field with the value "good" and the comment with "working fine", for super-drawer "LBA41":

```
UPDATE tileonlinestatus SET team4='good' , team4comments='working fine'  WHERE
drawer='LBA41';
```

3 Front-end application

Following the requierements, the central web services hosted by CERN has PHP installed with version grater than 4.X with GD and MySQL extensions installed. The Tile Online Status browser application will be able to run hosted by CERN's central web services.

A web interface is the most compatible interface to share information among multiple users. Thereby, the Tile Online Status browser application uses a web interface to display this information.

The Tile Online Status front end application is then written in PHP as it matches the requirements to be able to connect to a relational database and draw images with fairly little effort.

Nevertheless, there is still a need for a dedicated development to draw the TileCal barrels in a fairly easy way for the future developers. A TileCalBarrel PHP class, will be described in Section 3.1, as being a tool to draw one barrel divided in wedges and each wedge capable of supporting various layers. Also this class allows the user to fill the drawing with colours as specified in the requierements.

Three diferent web pages will be described. An index view of the Tile Online Status, grouping all information from the four barrels in Section 3.3. A table view resuming the same information without the barrel drawings in Section 3.5 and the update status page, where the user will be able to update the status of any of the groups and add any comment in Section 3.4.

3.1 TileCalBarrel class

The TileCalBarrel class is a PHP class used to draw TileCal like barrels. This class requires PHP version grater than 4.X and the GD extension for PHP installed. It doesn't matter if the GD extension is loaded by default because it can be dinamically loaded at runtime. It should be loaded before the declaration of this class.

The colors used for the TileCalBarrel drawings are defined as variables of the class. The variable name and the RGB color code is presented in table 5.

Variable	RGB code
white	0xFFFFFF
green	0x00FF00
blue	0x0000FF
red	0xFF0000
yellow	0xFFFF00
orange	0xFF8000
black	0x000000

Table 5: Color variable names VS RGB codes.

The TileCalBarrel main methods will be described. This aims to be a helpfile for any further developments that want to use this TileCalBarrel class.

3.1.1 TileCalBarrel ()

This is the public constructor of the class. Initializes internal memory and the colors. No colors can be used from a non-initialized class.

3.1.2 Function SetLegend()

This function is to set the legend of the barrel in the corner of the picture. Namely LBA00, LBC00, EBA00, EBC00.

3.1.3 Function SetName(string name)

This function defines the name of the barrel. It can only accept values LBA, EBA, LBC or EBC. Initilizes internal memory and attributes that define the direction of the rotation for the numbering of the drawers inside the barrel.

3.1.4 Function SetSectorLabels()

This function sets the sector labels to be visible. If not called, the sector labels are not visible. The sectors names are not user-defined but defined by the name of the barrel.

3.1.5 Function SetNumLayers(int numlayers)

This function sets the number of layers in which to divide each super-drawer. The numlayers parameter is a numeric value that can handle any number between 1 and 9.

3.1.6 Function SetModuleColor(string module, int layer, int color)

This function defines a certain color for a given module and layer. The module parameter is a numeric value from 1 to 64. The layer is a numeric value from zero to the maximum number of layers minus one. This value is defined by function described in Section 3.1.5. The color is a color defined by the class that is initilized at the constructor.

```
$b->SetModuleColor(1,1,$b->green);
```

3.1.7 Function Commit()

This function commits all the information needed to draw the picture including the module colors. It is required to draw the picture.

3.1.8 Function Draw(string file = null)

This function draws the TileCalBarrel into a given file or to the stream. If file parameter is null the function will send x-png headers to the client to encode the picture as a image file.

3.2 Barrel drawer

The OnlineStatus.php file is the TileCalBarrel drawer file. This script uses the TileCalBarrel class to draw a TileCal barrel. The information is extracted from the relational database and the picture is thrown through the main stream to the user client browser. Thus this file when openned from a web browser will popup a image file.

Following there is the example how to use the TileCalBarrel class correctly. This is done from a php script including the TileCalBarrel class file.

```
Include ("TileCalBarrel.php");

//Create an instance of the TileCalBarrel

$barrel = new TileCalBarrel();

//Set the barrel name. Initializes internal properties

$barrel->SetName("EBA");

//Define the number of wedges you want for your barrel

$barrel->SetNumLayers(4);

//Set if you want to see the Sector Labels

$barrel->SetSectorLabels();

//Set the barrel Legend depending on name

$barrel->SetLegend();

//Commit all the chages before drawing

$barrel->Commit();

//DRAW
```

```
$barrel->Draw();
```

3.3 Main display

The TileCal Online Status main display is organized in a two column table as shown in Figure 1. The upper row displays the LBA and LBC barrels and the lower row diplays the EBA and EBC barrels respectively. The representations of each of the barrels is done in a so called "Rik's" representation where the barrel is divided into 64 wedge and each wegde is divieded into 4 layers corresponding to each of the four commissioning teams: LV, HV, Team4 and Team5.

The display of each barrel is done through the TileCalBarrel PHP class embeded into the barrel drawer page described in the previous Section [Section 3.2].

The central space is reserved for a legend including the layers definition and color properties. From top to bottom, the first item is the layers definition, second the color legend and the third the last update date that gets updated each time a user updates the status for any super-drawer.

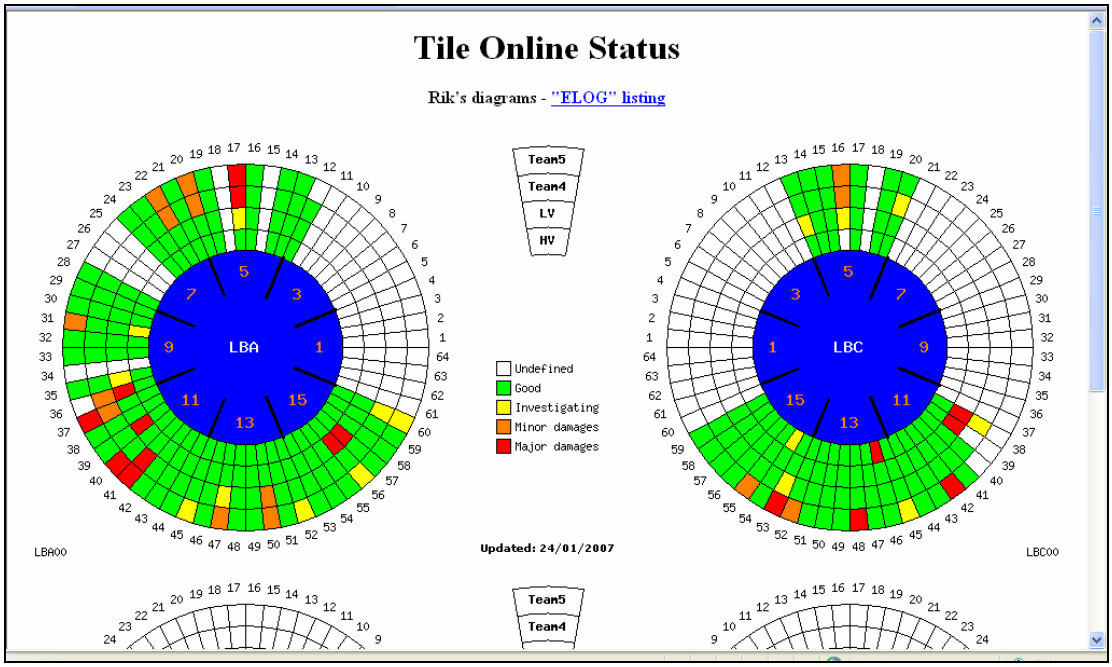

Figure 1: Main display web page.

3.4 Status updater

By clicking on a module from any barrel, the user enters a status updater page. The Status of each of the single commissioning team can be changed and a comment appended. Finally, by clicking update, the values will be stored to the database.

To return to the main display, the user has to click the index button.

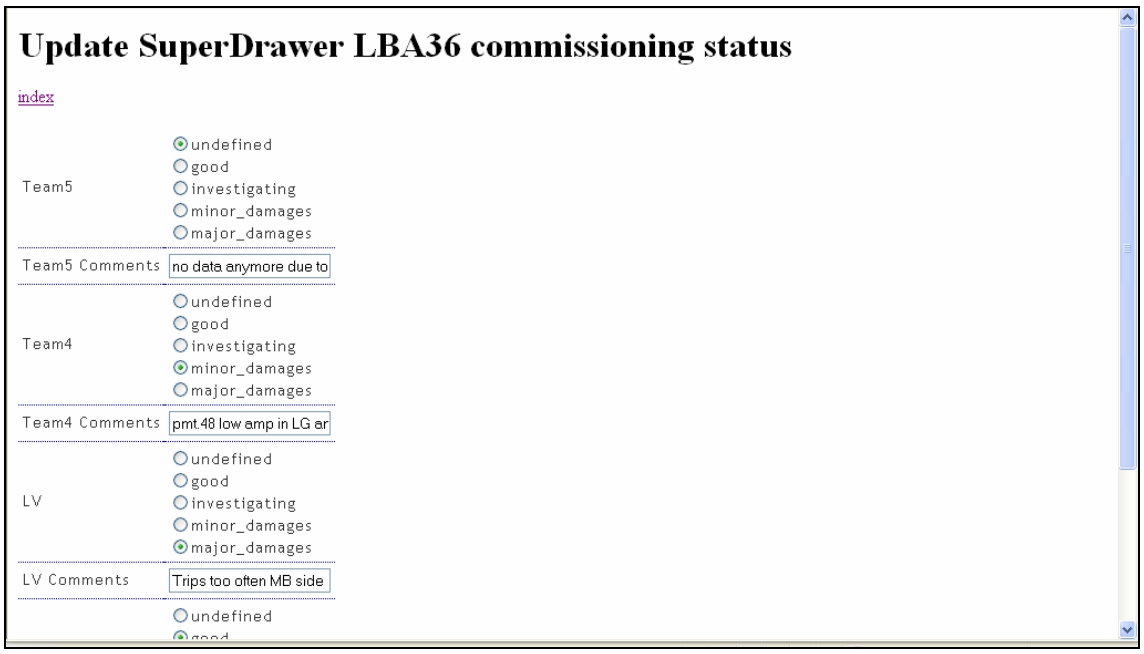

Figure 2: Status changer web page.

3.5 Table viewer

The whole listing of comments can be viewed by the table viewer, as shown in figure 3. To access the table viewer, the user has to click on the ELOG listing hiperlink from the main display page [Section 3.3].

The user can then choose between the complete or the zero suppresesd listing. The zero suppressed listing will remove all the rows where there are no commnets.

To return to the index page the user has to click on the "Rik's diagrams" hiperlink.

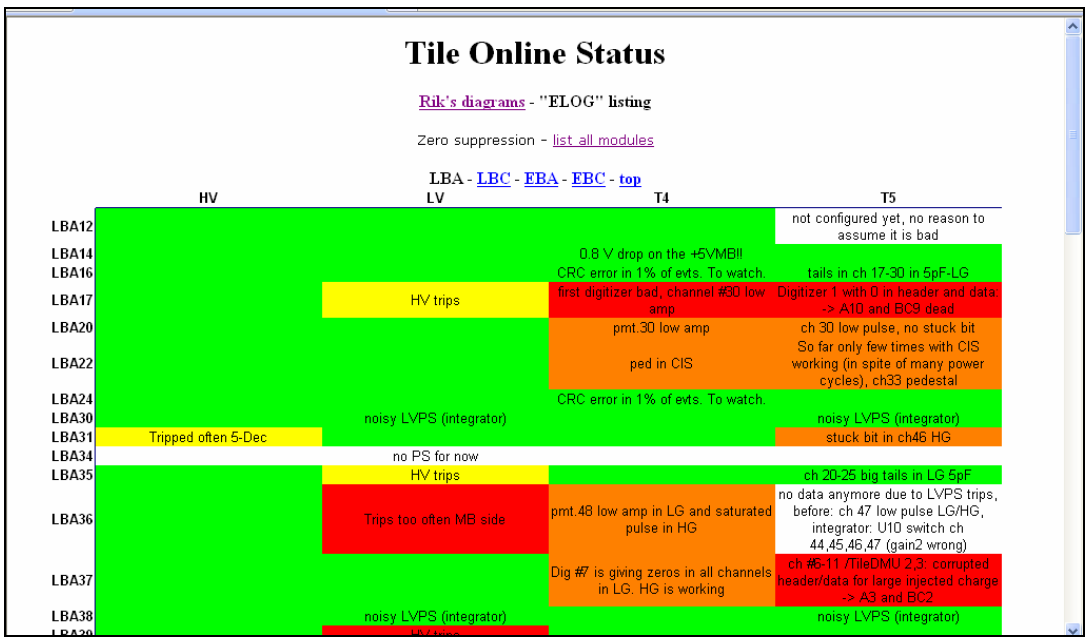

Figure 3: Table viewer web page.

4 Software release

The Tile Online Status browser application is distributed as a CMT package belonging to the Tile Online CVS repository. All the code can be found in the TileOnlineStatus CMT package.

5 Conclusions

At daily life of the commissioning of the TileCal super-drawers, it is useful to have an idea of what is the status for all the commissioning teams. This tool allows the people involved in the commissioning to have this general idea at a glance. Through the main display [Section 3.3] of the Tile Online Status browser application, the user can view the global status of the commissioning teams in an easy way to undestand.

The Tile Online Status browser application is accessible through the internet. This allows everyone to have a general idea of what is the status of the commissioning of the TileCal super-drawers.

Furthermore, the Tile Online Status browser application can be used to generate figures for presentations on the status of the commissioning of the super-drawers.

Very little effort has been put into the development of this tool, what makes the Tile Online Status browser application easy to modify and deploy for other purposes.

6 Acknowledgements

Carlos Solans wants to aknowledge the help of the people without whom this Tile Online Status browser application will not be real: specially Oleg Solov'yanov and Rik Yoshida from whom "I put together the ideas they suggested".