**LENSVIEW SOFTWARE STRUCTURE DOCUMENT**

This document is meant to better understand the structure and design of the LensView software before browsing thru the source code. So I suggest a reading of this document before opening the source code or else the entire source code will appear like greek and latin !!

Okay to start with the LensView software is a photographic sales software. The GUI will look almost identical to <http://www.artleaves.net> except for the corners of the screen where there are empty spaces which will be filled with advertisement panels to play advertisements. More on this later in the document. The whole idea is to email customers with a link which when clicked will copy and self install the software on the users computer. There will be a readymade install available for HOME users and there will also be a custom install also referred to as self install.

The basic structure of the LensView software is as follows

A very crucial, essential and critical Mains Power Supply Board called ‘class LiveDB.java’. This class is nearly 99 % static in content but it contains all the major global variable declarations essential for the application to function and also some Generic Wrapper methods which are very frequently used through out the source code. The Mains Board also contains tables and indexes.

The Mains Board also contains LiveSharedDB which is the actual database of tables only.

These are tables zero thru seventeen which are called real tables with real data in serialized files.

This LensView application is such where each and every new user is running his own instance of the

Application with his own snapshot of LiveSharedDB which is kept consistent.

For a detailed listing of these tables in LiveSharedDB look up the following source codes.

ArtLeaves.java -- Basic Installation Information also very critical since it contains email id’s

Without which the application will malfunction. The application uses emails

Very extensively to email log files, taking backups, system administration etc.

CelebrityTable.java -- This lists the Celebrities that you would like to honor with special status.

ChangeTracker.java -- This is a very critical table that keeps a backup of each and every transaction

In the database Updates, Inserts and Deletes. This table is used for recovery

At very many places in the Application.

CrashRecovery.java -- The above table keep records of all transactions. But this tables is an addition

That records just the last transaction in the database for crash recovery in

Case of power failures etc.

CustomerRegister.java -- This maintains all customer details

DefaultAdvertisingImages.java -- This maintains details and locations of files for Advertisements which

You want played/displayed when the application is running.

FileTableDataFileMap -- This is a very critical Master without which the database will not function.

In fact at install time this is the very first table to be created.

InstalledTables.java -- This table is a Master for all Installed tables in the database which is this list.

It will also contain an entry for Itself.

LanguageSupport.java -- Which are the Languages supported by LensView ? You may support a few

Languages and some other person installing this software may support other

Languages of his choice.

LanguageTitles.java -- This contains transalations from English to various languages for small

language labels on the screen not exceeding two words.

LanguageTransalations.java -- This contains a language file name which contains the transalated

passages in several languages for each of the descriptive write up

that appears when you click on a picture at artleaves.net

PaymentType.java -- What payment types such as Credit Card, Cash On Delivery, PayPal etc

PictureDatabase.java -- This is the MASTER for all the pictures you wish to sell. Must contain at

Least one picture entry.

PictureOptions.java -- This is the MASTER for all picture options that come up when you click

On a picture at artleaves.net such as Framing Options.

PlatformClearance.java -- This table is very crucial/critical for the Database Engine to work. It queue’s

all requests for the database engine reads and writes and then the Database

Engine picks one entry at a time from this Queue and executes requests.

It also helps ensure that the database remains consistent and your snapshot

Of the database also remains consistent.

ReportsTable.java – Self Explanatory but this table is populated by SortandIndex.java thread

SearchDatabase.java – User search feature is restricted to the words in this database

ShoppingCart.java – Contains all information relating to user shopping. The application cookies

The main program is contained in LensView.java. It also contains code for starting the application both

As command line application and an applet. Other than this there is also InstallDB.java. This contains code for Installation of LensView. It also contains very critical methods used during initial startup of the application every time the user starts his own instance of the application. It also contains important methods like fsck (). I would suggest that you go thru all methods in both these source files to understand the logical flow during initial startup.

The LensView software is multi threaded and the following are all the threads in the system.

Most of these threads are daemons that keep running even after the user exits the application.

The daemons automatically finish pending tasks mostly reports and exit smoothly.

AutoBackup.java – As the name suggests backs up the entire database. Most of the backup is done to a

mailbox that you specified at Install Time in table ArtLeaves

BillionHearts.java – Reporting Thread

IntranetCompliantRadioTaggedCollar.java – This is the Data Load and Reload Thread

LaunchGUI.java – This is the GUI thread very complicated

RefreshDatabase.java – This thread refreshes your snapshot of database periodically

And also has a static method for indexing of tables.

RoyaltyTracker.java -- Reporting Thread

SortandIndex.java – This thread was originally meant to sort as well as index but the indexing

Functionality has been moved to RefreshDatabase.java. What this thread

Does is to check how many users have logged in and allocate reports generation

Tasks to each application user by Inserting records in ReportsTable. I also

Creates some sort tables at the very start which are not fully used in the

Source code but nevertheless they are usefull sort tables that are available

For use to anbody who would want to modify the source code.

Spy.java – As the name suggests this thread has methods to keep an eye on possible intruders

or hackers and issues bells/whistles/alerts in the LOGFILE. It also has very important

dump () methods to dump the In memory LiveSharedDB contents of several tables to

disk as text files separated by field separator which can be used to diagnose and also

reload the good data only.

SystemExit.java – This thread will do a System Exit of the smooth exit kinds and it will systematically

Serialize all tables to disk along with some clean up procedures before exit.

ThreadController.java – This is a critical component that controls all the threads using different

Transmissions. Once this thread is running it is self sufficient and takes

Total control of your application.

WriteTables.java – This is the fast commit thread that checks every now and then and serializes tables

that have changed to disk.

ERROR LOGGING : Take a look at method grassHopper () in class LiveDB for all details of error logging.

This method produces an entire A4 sheet horizontal format with all the system information needed

For that particular error so as to catch even a intruder on the system immediately. It also has sounds

Like bells and alarms that sound every time an error is logged.

Also take a look at class LiveDB method redirectstderr ().

Also take a look at class LiveDB variable ErrorLogFile which contain the log file name for the current

User session. The log file name changes for every user session and will be automatically emailed to

The installer of the software. The reason for this is we don’t tampering of the log file. And the filename

For the log file is very verbose and very different for every user. So it is difficult to tamper.

REPORTING : All reporting is done using the GenericReportWriter.java

This is a highly flexible and very versatile report writer that is also intelligent.

THE ADVERTISEMENT PANELS : Look at NorthAdvertPanel.java, NorthEastAdvertPanel.java, NorthWestAdvertPanel.java, SouthAdvertPanel.java, SouthEastAdvertPanel.java, SouthWestAdvertPanel.java, EastAdvertPanel.java, WestAdvertPanel.java

Before you take a look at each of the panels code take a look at LaunchGUI.java and

FirstPage.java. Look at method PlayAllAdverts () in FirstPage.java and understand it well.

API’s to USE : This source code for LensView software extensively uses and is totally dependent on the following API’s. Preferably get the cross platform versions of these API’s free downloads and install the same along with NetBeans IDE 6.9.1 or higher. You may alternatively use the platform specific API’s freely downloadable.

Java Mail API

Java Sound API

Java Media Frameworks

I hope this suffices to get started with opening the source files and reading the method documentations which are very detailed in documenting changes and corrections made to the code.