**HOW TO INSTALL THIS SOFTWARE ?**

The Name of this Software is LensView (Photographic Sales Software). You can change the

name but that is all advanced stuff where you will have to change source code & recompile.

Okay there are three ways to install this software from the EndUser viewpoint.

From you viewpoint of setting up stuff on your server it is detailed later.

1. Choose the pre-installed web start at <http://www.fotokrafts.net/>fotokrafts\_java.html

But this is only for HOME EDITION USERS. But before you click set your computer name

to Zoa. It should work even with a different pretty name but why take chances. Make

this a Windows version by using UNC for path names during data-entry. It should work

just fine on non-Windows boxes but I expect some glitches so add Option4 for Unix like

boxes like Mac,Solaris etc.

1. Choose the self-install web start at <http://www.fotokrafts.net/fotokrafts_java.html>

This is for any kind of user. But before you click set your computer name to Zoa.

It should work even with a different pretty name but why take chances. If you wish to

Use this option for a PRO-INSTALL you will have to trick the LensView application and

slip in a LENSVIEWPROPERTIESFILE.txt with lensview.homedition=false. You can manage

that if you want to !!!

1. Download the zip file ArtLeaves.zip at <http://www.fotokrafts.net/fotokrafts_java.html>

Unzip zip file on your computer. Four Packages will be present default, MaintainData, PitStart,

DataEntryPrograms. Default package will not show as a directory on disk. It will show on disk as

MyJavaProjects/ArtLeaves/src/\*.\*. This default package will contain some .png images, \*.mf files

for manifests, web.xml, \*.jnlp for the webstart to be installed on your TOMCAT.

If user chooses option 1 they have nothing else to do beyond a mouse click. The application starts

running on that computer. All that you have to do is to email a URL link pointing to your webstart

for option1 to all prospective customers (home users) and they start using LensView immediately.

Therefore ensuring immediate orders.

If option 2 is chosen then a self install can be done by user as HOME EDITION OR PROFESSIONAL

EDITION. Even with this option there is very little to do it will go thru the installation process like a

breeze. Even HOME\_EDITION will be set by a properties file (LENSVIEWPROPERTIESFILE.txt)

parameter. You may choose option2 for PRO INTRANET install after tweaking file

LENSVIEWPROPERTIESFILE.txt and setting key lensview.homedition=false. If you choose option 2

for an install you will want to tweak the properties file a bit here and there

The properties file I am talking about is the LENSVIEWPROPERTIESFILE.txt. Don’t copy it elsewhere

The application will do it for you. Tweak it in the same directory. Okay the properties file looks like

this in next page. This LENSVIEWPROPERTIESFILE.txt will also be copied to all user home directories

automatically. After Installation you will find master in conf subdir LENSVIEWPROPERTIESMASTER.txt

THIS MASTER FILE is where the SYSADM must make his periodic changes. There is another file that

Is also automatically copied to every users home directory called photodbapp.conf. You cant tweak

this file post installation. The photodbapp.conf is a critical file and a backup will be present in the

subdirectory conf post installation with file name photodbapp.conf.dsk.bkp.

**Changing Property key=value pairs**

Out of these you might change autoinstall to false to ensure a CUSTOM INSTALL. Also set values

for copyimagesfrom and copyimagesto if you set autoinstall to false. Also set transfilesfrom to a

directory if you set autoinstall to false.

Also tweak the homedition to true or false as you wish. If false it is PROFESSIONAL EDITION. This

false will be the default value if you leave homedition blank without setting any value.

Now tweak STORE SETTINGS SECTION OF THE FILE storeholdays, otherholidays, storetimings.

Next is the DAILY TEMPERATURE SECTION set cloudy and rainy to true or false (Peep out of window).

Also set temperature by winking at some pretty at your office.

Next is the CURRENCY section which is almost a daily grind except for one setting currencythreshold

If you feel that the USD is going to get weaker than other currencies in the immediate future then set

a value lower than 0.41. If you feel that other currencies like pound, euro, rupee are going to get

weaker to the USD in the immediate future set a value higher than 0.41. There is a default of 0.41

in the application that should work okay for quite some time. This is just a safeguard feature to

prevent accidental tweaking of the currencies exchange rates in the properties file either accidentally

or by a prankster.

Within the CURRENCY section there are three more keys to set namely eurorate, inrrate, usdrate,

poundsterling. Google any currency calculator every week and set these values. You can also do it

on a daily basis.

#INSTALL SETTINGS

lensview.autoinstall=true

#EDITION SETTING

lensview.homedition=true

!LANGUAGE SETTINGS ALSO AN INSTALL SETTING

# End this always with a comma specify only two two letter language codes other than en,ta,sw

lensview.languageselections:fr,de,

# no defaults have been defined for these keys unlike all the below that have defaults

# I have commented these two lines below fill up with appropriate values on you machine

# copyimagesfrom is to specify the directory of images for Pictures and all advertising images

#define the from directory for PRO installs only. For Home Edition a directory called imagery will

#be created automatically under users home directory for windows and under user directory for

# non-windows. During testing for HOME edition create a directory called “imagery” under user

# home directory for windows and under user current working directory for non-windows and

# copy all picture images and default advertising images to this directory

! copyimagesto is to specify the directory to copy to the contents of the from directory

#lensview.copyimagesfrom:C:\\WINDOWS

#lensview.copyimagesto:F:\\PHOTOAPP

! This key value pair will contain directory location of transalated language files on disk

# This will be copied to etc directory by LensView application

# make sure your narratives file gui\_text\_area\_type JNarrative is also a path pointing to

# LiveDB.serializedfilespath/etc directory.

# I have commented the line fill up with appropriate values on your machine

#lensview.transfilesfrom C:\\TRANSFILES

! STORE SETTINGS

lensview.storeholdays Sunday,LaborDay,Thanksgiving,Christmas,Good Friday, \

Mandatory National Holidays

lensview.otherholidays Deepavali,Ganesh Chaturthi,ShivaRatri,Janmashtami,\

Buddha Purnima,Bakrid,Hanukah

lensview.storetimings=10 AM to 5 PM LOCAL TIME

# DAILY TEMPERATURE SETTINGS

lensview.cloudy :false

lensview.rainy :false

lensview.temperature:28.00C

! CURRENCY SETTINGS

!you may uncomment then next line

!lensview.currencythreshold=0.41

# OR you may define it as "" this way there is a default in the application code

lensview.currencythreshold

# DAILY TWEAKS TO CURRENCY RATES

! HOW MANY EUROS MAKE A DOLLAR

lensview.eurorate=0.75

! HOW MANY RUPEES MAKE A DOLLAR

lensview.inrrate :49.16

# HOW MANY POUNDSTERLING MAKE 1 DOLLAR

lensview.poundsterling 0.65

! HOW MANY USD MAKE 1 EURO

lensview.usdrate:1.33

NOTE:- This lensview.transfiles key=value setting must contain as value the name of the directory

where you have placed all your transalated language files and narratives files to ensure a bulk copy

to the etc directory. The narratives file for different languages do not follow any naming convention

but it is better that you follow any naming pattern and adhere to it. These narrative files must not be

html files. Just save them as text files. In the etc subdirectoy post install you will find a file named

unavailable.txt don’t erase that.

Also it would be better if you the VENDOR COMPANY tweak this LENSVIEWPROPERTIESFILE.txt

and email it to the system administrator at each company where lensview software is installed.

Sysadm can then copy file at local installation every morning to LENSVIEWPROPERTIESMASTER.txt.

In the next release of the LensView application you can email sysadm commands of LensView to a

specific email Id and we will download that email from the mailbox and then run the command at that

installation. So your dependence on the sysadm will reduce to a bare minimal.

Choose option 3 for a Real Guru Professional install only where you feel the need to tweak the

Source code and re-compile the distrib to your needs. Mostly you will need to change the name

Zoa in the VersionControl.java file to your server name and also change the application name if

You so desire to confuse hackers a wee bit. This will work with any pretty name other than Zoa.

You may choose option 3 for an INTRANET installation (PROFESSIONAL EDITION) or even a

HOME EDITION but it does not make sense with a HOME EDITION unless you are learning Java.

**For Option three look at COMPILING THE PROGRAM Section**

**POST INSTALL DIRECTORY STRUCTURE**

Post Install look for a top level directory on your disk. If you set lensview.autoinstall=true in your

LENSVIEWPROPERTIESMASTER.txt (at install time) then a top level directory by the name of LensView

Would have been created on your disk. In the case of lensview.homediton=true it will be created on

your C: drive (IF O/S is windows) and In the case of lensview.homediton=false networked drive letter

(If O/S is windows). In the case of unix it will be created under your cwd from which you installed.

If you set lensview.autoinstall=false in your LENSVIEWPROPERTIESFILE.txt (at install time) then a

top level directory by the name you chose exists on the drive/partition you chose.

Under the top level directory you will find the following directory structure created

\* SUB DIRECTORY NAME CRITICAL PURPOSE

\* etc no misc files that dont fit in elsewhere like shell scripts, fsck

\* spool no print requests for a later time (next release)

\* archive no archive database files

\* bin no supplementary programs other than InstallDB

\* tmp no temp files if for some reason system tmp cannot be used

\* var no files with a short life

\* dump no dump files

\* dataload no data entry files to load including corrected dump files

\* syc no ipcs synchronization files

\* daemon no daemon files

\* policy yes policy files. Yes you find LensView policy file PhotoDBApp.policy

Post install. Take a look at it but don’t tamper with it unless you

are a PRO. If there is a LiveDB.USRHOMEDIR + ".java.policy" file

consider deleting that file after adding some of its contents to

the LensView application policy file.

\* conf yes configuration files

\* reports yes all reports generated

\* images yes all audio, video, photo images including companylogo

\* version yes version control files

\* usr yes user login info and other user related files

\* backup yes auto backups, restore files, old backup files

\* rowsets yes next release will contain row sets files for all tables

\* dev yes files for some device names (next release)

\* libs yes next release for some libraries if needed

\* dbf yes database files Integrated release integrating other databases

NANPA, SINDPA, JANPA etc

\* sccs yes source code control system (next release)

**SERVICE PATCHES (UPDATE DAEMON)**

This update daemon will be emailed to home users. When they click on it a program will run on their

computer that will download and copy some new images to their disk and also replace their

LENSVIEWPROPERTIESMASTER.txt This will be made available after the Beta Release of the software.

But this is optional. Anyways you don’t have to worry about the LENSVIEWPROPERTIESMASTER.txt

really because the final invoicing is always done in US Dollars. So at worst what might happen is that

the home user has outdated currency exchange rates for all international currencies which may result

in flawed currency conversions when the home user switches currencies. It can also replace the

executable on your system with new version or release. A new version post BETA will be service patch.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

Lotsa Stuff including language transalations and data entry files which you re-use at other corporate

INTRANET PROFESSIONAL installations around the world.

First start with language transalaltions. Use a language transalator for very many languages available

At <http://www.fotokrafts.net>. On opening this site you will find several lines in several languages

Scrolling horizontal as marquee at bottom of page. Click on any of those and it will open a page. AT

The bottom of that transalated language page will be a link to the language transalation machine.

Bookmark that into your favorites.

If website is down try this URL link <http://www.translation-guide.com/free_online_translators.php>

If you look at the bottom pane at <http://www.artleaves.net> it will look like this image below

bottompane.jpg

When the user clicks on any of the titles the application should display an html page containing the

appropriate language transalation based on the language selected by the user.

So for example “Contact Us” . If user clicks then based on the currently selected language he is served

a html file in that specific language. So for every language that you chose to install on LensView you need to create a language file in html. That filename must be LENSVIEW + “Contact” + “\_” + “Us” + “fr” + “.html”=LENSVIEWContact\_Usfr.html. Wherever there are one or more spaces between words replace with underscore ‘\_’ character. In the above filename fr means French. Use two letter iso language codes google the list on the internet. Look at LANGUAGE SUPPORT SECTION OF THIS DOCUMENT ALSO

OR try <http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes>

There are one more set of language files that you need to create for the opening page with lots of text

In the center window at <http://www.artleaves.net> with the maple leaf image. That page needs file names in this format for each of the languages you choose to install on LensView Format for language file names is "LENSVIEWFIRSTPAGE" + “fr” + ".html" = LENSVIEWFIRSTPAGEfr.html

The LensView Application is an intelligent application which looks for these filenames and if it does not

Find that file on disk in the etc sub directory it serves the user with a default English file. So htmlfiles for language code “en” must exist on the disk. The LensView Application cannot check for this and does not.

Now all these language transalation files need to be copied to any directory on any drive accessible over the local network. So if you choose to create these files on H:\\TRANSFILES then you need to tweak the

LENSVIEWPROPERTIESFILE.txt in your USER DIR (cwd) before starting the install process for PRO EDITION

Uncomment this line in the LENSVIEWPROPERTIESFILE.txt and replace the ‘C’ drive letter with any network UNC path on your server (if Windows) or else a unix path such as /home/user/pskris

#lensview.transfilesfrom C:\\TRANSFILES

So the application will read this key=value pair and copy language transalation files from the directory

You specified to the etc subdirectory during installation.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

**DATA ENTRY**

**Note :- For Pre-Install version Option 1 all this data entry would have been done for user and data entry files will be copied to his computer automatically. Even for Option 2 the data entry files will be autocopied to users computer. This passage below details what you the vendor company need to do to make the user experience a real breeze.**

After transalating language files and placing them in the appropriate directory (optional step as I mentioned above) launch your DataEntryProgram package executable or jar whatever you name it.

Before you compile this program change MaintainData.LiveDB.fieldseperator to whatever you want.

Stick to the default ‘|’ as it is the best.

Start the Data Entry for each of the tables in that program. You will notice that the table choices are

Listed in groups. Take one group each for data entry. It is a multi-user program for quick data entry.

The Data Entry Screens are fairly verbose with tool tips and validations for form fields at most places.

Look at the underlying disk and you find the character data files for all tables in a directory called LensViewDataEntry. Copy this directory and all its contents to a pen drive (It will be usefull later).

Each of data entry files on disk will have the name as “tablename” + “\_de” + “.de”. So if table is filetabledatafilemap the data entry filename will be filetabledatafilemap\_de.de . The tablename in the above file naming formula will be lowercase.

When you do your data Entry for table filetabledatafilemap enter the datafilename as the same path

LensViewDataEntry. For HOME\_EDITION pre-install as in Option1 or self-install as in Option2 use a relative path (mandatory). Relative paths must enter as LensViewDataEntry\filetabledatafilemap\_de.de This application will convert the relative path into an absolute path automatically. You don’t have to worry about that. This relative path LensViewDataEntry\filetabledatafilemap\_de.de is for Windows. For non-Windows the relative path will be LensViewDataEntry/filetabledatafilemap\_de.de . Remember all the other operating systems that LensView application supports are unix variants.

You must enter the path as relative for HOME\_EDITION in all other table fields such as

TABLE NAME FIELD NAME

PictureDatabase picture image location

FileTableDataFileMap datafilename

LanguageTransalations Language File Name

DefaultAdvertisingImages Directory Name

ReportsTable pathname

To create Java animations for LensView Application you might want to try DimensionX's The Easy Animator (TEA) (previously known as JAM). It is interactive and will let you create most animations and must be free by now. While doing data entry for LanguageTransalations table it will expect a Language File Name for every picture code for every language. Language file name is nothing but narration that appears with every picture in user chosen language. Use URL link <http://www.translation-guide.com/free_online_translators.php> and

save each of these narratives file as a text file.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

If you are doing a PRO INSTALL for **Hetrogenius Intranets** you must install it on a Windows server for now until we find some way to fix that glitch for you. As regards path names during data entry you must must use a Windows UNC path name which is in format [\\\\ServerName\ShareName\Resource](file:///\\\\ServerName\ShareName\Resource). Here Resource is a path relative to the share. So use LensViewDataEntry\filetabledatafilemap\_de.de . The data entry form does not validate for a UNC path specifically. So you need to keep this in mind during data entry. Just one thing this UNC path must be tested for error free network access before usage.

Use UNC path names if you are installing on a Windows Server only. If you are installing on Unix Box use absolute path names. We will convert it to a URI in the LensView application but there is a technical glitch to it because a URI on one instance of the JVM on one machine may not work on the other instance on the other machine perfectly at all times since they may not be able to resolve to the same absolute path.

In a professional install you may copy all contents of LensViewDataEntry to some other directory or directories on the same share and enter the UNC path name accordingly. So therefore your UNC path during dataentry may look like \\Cupertino\BayShare\ LensViewDataEntry\filetabledatafilemap\_de.de

OR \\Cupertino\BayShare\ DataLoad\filetabledatafilemap\_de.de

AND \\Cupertino\BayShare\ Adverts\defaultadvertisingimages\_de.de Get It !! ??

But for Gods Sake use the same share name for now since I expect some problems !! I will come up with

Some patches to the code in about ten days time from 16/10/2011.

Even if it is not a Hetrogenius Install at your location. If you are installing on a Windows Server use a UNC path always. Because later you may add unix boxes and switch to a hetrogenius intranet usage !

If you have to use LensView application on a hetrogenious intranet then you must launch it as an Applet

From any of your intranet pages. For Gods Sake don’t load it on the internet or else I am dead !!!

As I mentioned in one of the earlier extent LensView is intelligent enough to understand that some language specific data is missing in the database and defaults to using English as an action of last resort.

Even if you are lazy & enter only one set of data for language code “en” (English) in LanguageTitles table.

The Language titles table looks like this

private String langcode; /\* two letter ISO language code \*/

private String englishlabel; /\* The label in English \*/

private String language\_text; /\* Language Text in Unicode \*/

private long rowid; /\* 4 FIELDS \*/

The label in English is any label which you use on the GUI as at <http://www.artleaves.net>. Language text for just language code “en” could be the same as englishlabel or it could be Old English !!

So if you are somewhat lazy during data entry and enter just one set of data for “en” don’t worry about

Other language switches the LensView application will pick this default.

NOTE:- There is one table missing in the DataEntryPrograms which is the ArtLeaves table. This table will

Have to be entered by you using notepad and supplied to the user for all Option1, Option2, Option3.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

The table structure looks like this

private String compname ; /\* Company Name \*/

private String addr1 ;

private String addr2 ;

private String addr3 ;

private String addr4 ;

private String city ;

private String state ;

private String country ;

private String zipcode ;

private String email1 ; // Mandatory emailid From which all emails will be sent must be gmail id

private String email2 ; // Mandatory emailid to which LogFiles only will be emailed

private String email3 ; // Mandatory emailid to which PurchaseOrders only will be emailed

private String email4; // Mandatory email id which is notification mailbox telling you about email in

other mailboxes

private String email5; // Optional email id for Invoicing your customers otherwise notification mailbox

used

private String email6; // Optional email id for Return Material Authorizations otherwise notification

mailbox used

private String email7; // Optional email id for Legal Matters/Disputes/Correspondence otherwise email1

private String email8; // Optional email id for all other Reports otherwise notification mailbox

private String email9; // Optional email id for all In-Office Correspondence Personal .... !!! otherwise

notification mailbox

private String email10; // Optional email id for dump files only attachments so make it large otherwise

notification mailbox used

private String email11; // Optional email id for overflow condition in any of above mailboxes otherwise

Email1

private String email12; // Optional email id for Customer Correspondence regular non-Legal type

otherwise Email1

private String email13; // Mandatory email id for CrashRecovery of Business Emails. It is a Backup

Mailbox

private String phone1 ;

private String phone2 ;

private String mobile1 ;

private String mobile2 ;

private String fax1 ;

private String fax2 ;

private String superuserpassword;

private long rowid;

The record terminator is the LiveDB.fieldseperator you compiled LensView Application with followed by

New line or carriage return.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

**MAIL BOXES**

The LensView application uses mailboxes extensively. Some are mandatory as detailed above. But it is advisable to fill in different email ids for all thirteen mailboxes. The reasons are mail box overflow and you could also increase the number of retries by altering the source code. There is section below called

SOURCE TWEAKS read that to know what to tweak and where. Also you may choose to alter the source

code to add an extra mailing out carrier service other than gmail. Some extra mailboxes will do no harm.

Make sure all these email ids are functional and receiving and sending emails.

So if your company name is let us say BroadCom you might want to use broadcom1@gmail.com thru [broadcom13@gmail.com](mailto:broadcom13@gmail.com) easy to remember your mailboxes then.

Don’t fill in null values for the following fields

private String compname ; /\* Company Name \*/

private String addr1 ;

private String addr2 ;

private String addr3 ;

private String addr4 ;

private String city ;

private String state ;

private String country ;

private String zipcode ;

private String phone1 ;

private String phone2 ;

private String mobile1 ;

private String mobile2 ;

private String fax1 ;

private String fax2 ;

private String superuserpassword; // fill in any valid characters

private long rowid; // must be 1 (one) form does not validate (INTENTIONAL)

**LANGUAGE SUPPORT**

**Next Data Entry for LanguageSupport table.** The structure of the table is as under

private long lid; /\* language id superfluous but let it remain may be usefull in future \*/

private String language ; /\* ISO Language code two letter \*/

private String name; /\* display name of the Language \*/

private long rowid ; /\* 4 FIELDS \*/

The data entry for languagesupport table will prompt for language codes. Fill in all these two letter language codes for each of the languages. Don’t worry about Display Names Data Entry Programs will spoon feed you with values wherever possible. The lid for each language must be unique. The form does

not validate this lid. Also form does not validate (INTENTIONAL) Rowid : Sequential ascending rowid.

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

String message = "ENGLISH, TAMIL, SWAHILI ARE MANDATORY LANGUAGES" + '\n'

+ "IN ADDITION TO ABOVE YOUR DEFAULT LOCALE LANGUAGE WILL BE LOADED" + '\n'

+ "THREE LANGUAGES ARE ALREADY SUPPORTED BY DEFAULT" + '\n'

+ "SELECT TWO MORE LANGUAGES FROM THE LIST BELOW" + '\n'

+ "French(fr), German(de), Telegu(te), Spanish/Isabella(es)," + '\n'

+ "Portuguese(pt), Sindhi/Sangeetha(sd), Punjabi/Sonam(pa)" + '\n'

+ "Arabic(ar), Russian(ru), Chinese(zh), Italian(it)," + '\n'

+ "Malayalam/Asin(ml), Nepali(ne), Norwegian/Twinkle(no)," + '\n'

+ "Finnish(fi), Gujrati/Ameesha(gu)" + '\n'

Don’t bother about the names of pretty polly’s in the message above just stick to the two letter language codes. Swahili = sw, Tamil = ta since they are not detailed above and English = en.

I you want to support many more languages at a later date no harm in adding many more language codes to table (two letter codes for now) from <http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes>

But if you want the user to choose from this extended set of languages then you will also have to tweak the source code

At some places as detailed in the SOURCE CODE TWEAKS SECTION. Also you will have to add more transalation files

For each of the additional languages and also narratives files and also some additional Label in LanguageTitles table during

Data entry.

**Next Data Entry for SearchDatabase table**. The structure of the table is as under

private String keyword ; /\* Keyword for search \*/

private String pic ; /\* Picture Code \*/

private long rowid; /\* 3 FIELDS \*/

for every picture code there could be several keywords embedded in the narrative text for that picture.

These keywords could be common and repeated in several picture codes. The picture code itself would

Be a keyword for a search such as PIC01, PIC49 etc. So enter all these keywords using DataEntryProgram and also enter a picture code for each keyword. If the keyword repeats itself in many picture codes add a new record for each repetition of the keyword with a different picture code.

Rowid : Sequential ascending rowed form does not validate

**Next Data Entry for PaymentType table.** The structure of the table is as under

private long pmtid; /\* Payment ID \*/

private String pmt\_type; /\* Description like Cheque, CC, WUMT, Cash, COD etc \*/

private long rowid; /\* 3 FIELDS \*/

Enter a unique payment id in each record on the data entry form. The payment type is valid payment types that you want to offer the user of LensView application. Presently we don’t support a payment processor in this release. In the next release we might. Until the CC or Credit Card would mean invoicing using PayPal or PayAmerica. Rowid : Sequential ascending rowed form does not validate

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

Next Data Entry for PictureOptions table. The structure of the table is as under

private String langcode ;

private int opt\_id ; /\* Options ID such as Frame Color \*/

private int sub\_opt\_id ; /\* sub option id such as Gold, Silver etc \*/

private String lang\_sub\_text ; /\* Language sub text for each of the sub options \*/

private long rowid ; /\* 5 FIELDS \*/

If you open <http://www.artleaves.net> and click on any picture it takes you to a picture specific page that contains a larger image of the picture and a descriptive narrative text below and lots of Framing Options to choose from as Labels along with combo boxes by their side.The values contained in these combo boxes are the sub options and the Labels such as Frame Color, Miscellaneous are Options.

So before you do data entry for PictureOptions table write down on a A4 paper what options and sub options you want to offer the user. So the first record for each new lingcode would be

Opt\_id = 1 and sub\_opt\_id = 0 and lang\_sub\_text using language transalator cut and paste then

Opt\_id = 1 and sub\_opt\_id’s = 1,2,3,4,n and the lang\_sub\_text using language transalator cut and paste

Rowid : Sequential ascending rowed form does not validate (INTENTIONAL)

**Next Data Entry for LanguageTransalations table.** The structure of the table is as under

private long lid; /\* Language id \*/

private String pic; /\* Picture Code \*/

private long gui\_text\_area\_no; /\*GUI Text Area Number NOT USING THIS KEEP FOR FUTURE \*/

private String gui\_text\_area\_type; /\* GUI Text Area Type -- TextBox etc \*/

private String lfn ; /\* Language File Name ready to serve \*/

private long rowid; /\* 6 FIELDS \*/

The lid here must be the same as the one entered in the LanguageSupport table. This Language

Transalations table is used only to fetch the Language file name which is the path and filename for

The narrative text that appears along with every picture when you click on the picture. So for every

Language id in the LanguageSupport table and each picture code there must be one or more language

file names that contain a transalated text passage. For this release just make it one language file name

for each lid, pic with gui\_text\_area\_no = 10 and gui\_text\_area\_type as “JNarrative”.

Rowid : Sequential ascending rowed form does not validate (INTENTIONAL)

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

**Next Data Entry for FileTableDataFileMap table.** The structure of the table is as under

private long tableno ; /\* store it here also for convenient programming \*/

private String datafilename; // the comma seperated notepad data filename with path

private String filename; /\* the persistent storage file name for table without pathname \*/

private String table\_identifier\_name; /\* Table identifier name LiveDB.tablenameidentifier \*/

private String classname; /\* Class Name associated with this Table \*/

private String table\_lock; /\* The Lock status for this table

\* table\_lock = "L" table locked by other process

\* table\_lock = "U" table unlocked.

\* table\_lock = "E" Emergency Declared

\* table\_lock = "POTA" Print Non-Bailable Arrest Warrants

\* for terrorist customers and also that bugar on the console.

\* Wait for next release no time now. There will be another

\* thread called SystemCop which will print Arrest Warrants

\* "You are Served" "Volley Will Follow".

\*/

private long pid ; /\* internal process id that locked this table \*/

private long rowid; /\* 8 FIELDS \*/

Actually this should be the very first table you must pick for data entry without which LensView application cannot install or launch. This MASTER table is a critical table that must only contain entries from tableno zero thru LiveDB.REALTABLES which is 17. While doing data entry the tableno’s must be entered in ascending sequence. That is 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17.

Datafilename is the path + filename in UNC format if you are installing on a Windows server otherwise on non-windows boxes start with “/”. Read the previous passages on entering path names in this document for clarity.

Filename : just enter any unique filename without any extensions. The application will plug in extension

.ser for every filename.

table\_identifier\_name and classname will be served to you stick with that don’t change it. I have it in a

text field because if you make changes to the source code you may need to alter values for these fields.

table\_lock must be “U” stick to this value. I have it in a text field because there may be some system crashes where you need to change this value.

Pid : the process id stick to value 0. I have it in a text field becase there may be some system crashes where you may need to change this value to up the database (extremely rare).

Rowid : Sequential ascending rowid. form does not validate (INTENTIONAL)

**All other tables Data Entry does not need explanation adequate tool tips have been provided.**

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

**SCANNING IMAGES**

All your images Pictures for Sale as well as advertising images must be scanned to 150 dpi magazine print. Try running LensView application with this scan. If it is fast in drawing these images and image quality is good. Then you can try scanning upto 200 dpi but not beyond or else the application will take

Time to render each of these images from a large file. Try even a newspaper scan if the application is slow.

**COPY FILES TO TOMCAT**

**But before that a little bit of editing**

Under MyJavaProjects/ArtLeaves/src you will find two jnlp files which are launch files. Open the files in an editor of your choice and save them back as text files with the same filename and extension.

In both files you will find localhost:8080 replace these with your server details and save the files. Don’t change the names of the JAR files stick with the default unless you want to change those names. To prepare an application for delivery you package it in one or more JAR files. Then you prepare a descriptor file in Java Network Launch Protocol (JNLP) format and place these files on the web server.

If you use Tomcat change the codebase entry in LensView.jnlp where there is this line which you have already changed (localhost:8080 to your settings). <http://localhost:8080/LensView/>. Now replace LensView with tomcat/webapps/LensView. Similarly in the other PitStart.jnlp file replace LensView/PitStart in line localhost:8080/LensView/PitStart/ with tomcat/webapps/LensView/PitStart

You can seal a java package to ensure that no further classes are added to it. So if you open the manifest files MyJavaProjects/ArtLeaves/src/PitStart.mf and MyJavaProjects/ArtLeaves/src/LensView.mf you will find a line called Sealed:true delete that line from the .mf files and save the files as plain text files with the same extension. Alternatively you can alter that line to Sealed:false and save the .mf files.

To experiment with Java Web Start you want to install TomCat from <http://jakarta.apache.org/tomcat> . Tomcat is a container for servlets and JSP pages but it also serves web pages. It is pre-configured to serve the correct MIME type for JNLP files.

You need to ensure that your web server reports a MIME type of application/x-java-jnlp-file for files with extension .jnlp (Browsers use the MIME type to determine which helper application to launch). This MIME type is associated with the javaws application. If you installed the JDK, theconfigration should be automatic.

If you use TOMCAT make a directory tomcat/webapps/LensView/WEB-INF and copy the web.xml file there. This directory is for Option1. Similarly make another directory for Option2 called LensView/PitStart/WEB-INF and copy the web.xml file there. Make sure both directories have read perms for all.

Now copy your LENSVIEWPROPERTIESFILE.txt after altering settings in it for Option1 and Option2 to

tomcat/webapps/LensView. Now change your LENSVIEWPROPERTIESFILE.txt once again for Option2 and copy to tomcat/webapps/LensView/PitStart .

**WHAT DO YOU THE VENDOR (COMPANY) DO TO INSTALL THE SOFTWARE ON YOUR TOMCAT ?**

**Continuation Page ………**

Next copy the entire data entry files directory by the name of LensViewDataEntry as is to each of the directories tomcat/webapps/LensView and tomcat/webapps/LensView/PitStart .

Now create two directories with same name “imagery” under each of the two directories tomcat/webapps/LensView and tomcat/webapps/LensView/PitStart and copy all your scanned images of pictures for sale as well as advertising images (including movies, audio, anmations) to each of the two directories tomcat/webapps/LensView/PitStart/imagery and tomcat/webapps/LensView/imagery.

Set read perms for all on the two directories called imagery.

Now write a .html file that points the browser to each of the .jnlp files. Save these .html files to the above two directories one for each Option. For example

Option1 the .html file will contain URL http://localhost:8080/tomcat/webapps/LensView/LensView.jnlp

For Option2 <http://localhost:8080/tomcat/webapps/LensView/PitStart/PitStart.jnlp>.

In both URL’s above replace localhost:8080 with your server details.

Next copy compiled JAR files LensView.jar and PitStart.jar to tomcat/webapps/LensView directory and

tomcat/webapps/LensView/PitStart directory.

Now there are four .png files in MyJavaProjects/ArtLeaves/src/\*.png

Copy /ArtLeaves/src/pitstart\*.png to tomcat/webapps/LensView/PitStart

Copy /ArtLeaves/src/LensView\*.png to tomcat/webapps/LensView

There is also another companylogo file in MyJavaProjects/ArtLeaves/src/companylogo.\*

Copy MyJavaProjects/ArtLeaves/src/companylogo.\* to /webapps/LensView/PitStart

Also copy to

Copy MyJavaProjects/ArtLeaves/src/companylogo.\* to /webapps/LensView

If the above stunts don’t work then you will have to sign your jar files and write your own authentication

Certificates. All this is very advanced so please contact a professional bugger for help. Try a large Pitza with pepperoni and extra cheese for a bribe !!.

**COMPILING THE PROGRAMS**

JAVA REQUIREMENTS : The LensView Application requires Java Mustang and higher releases for operation.

LIBRARIES REQUIRED FOR COMPILATION : JavaMail API (Cross Platform Version) and JMF API – Java Media Frameworks (Cross Platform Version). Even if the Cross Plarform Versions of these API’s are not available you can still manage. But you will have to compile a Windows Version and a Non-Windows version.

**Let us Start with the DataEntryPrograms for your use**. The easiest way to compile is using NetBeans IDE 6.9.1 and higher. Open NetBeans, right click on file DotingEntry.java under package DataEntryPrograms and choose Run File (Shift + F6). This is during testing.

**COMPILING THE PROGRAMS**

**Continuation Page ………**

To compile from the command line use the command javac [path]/DotingEntry.java

After compiling issue the command java DotingEntry

If there are error messages. Then set your PATH environment variable to find java and javac. The javac compiler is located in the bin directory below the installed Java Development Kit sotware directory.

Also check for access specifier public in DotingEntry.java class definition as public class DotingEntry.

Normally the java compiler places the class files in the same directory as the source files which can be a clutter. You can re-route the class files to another directory using the –d option of the javac command.

So cd to directory MyJavaProjects/ArtLeaves/src and issue the command

javac –d ../build/classes/DataEntryPrograms ./DataEntryPrograms

Create a temp file and add the line and save the file as a text file with name doting.mf

Main-Class: DataEntryPrograms.DotingEntry

To create a JAR file issue the command

jar cmf doting.mf DataEntry.jar

Now run the program by executing a command line as

java –jar /path/to/file/DataEntry.jar

**Now Start Compiling LensView application MaintainData**. The easiest way to compile is using NetBeans IDE 6.9.1 and higher. Open NetBeans, right click on file LensView.java under package DataEntryPrograms and choose Run File (Shift + F6). This is during testing.

To compile from the command line use the command javac [path]/LensView.java

After compiling issue the command java LensView

If there are error messages. Then set your PATH environment variable to find java and javac. The javac compiler is located in the bin directory below the installed Java Development Kit sotware directory.

Also check for access specifier public in LensView.java class definition as public class LensView.

Normally the java compiler places the class files in the same directory as the source files which can be a clutter. You can re-route the class files to another directory using the –d option of the javac command.

So cd to directory MyJavaProjects/ArtLeaves/src and issue the command

javac –d ../build/classes/MaintainData ./MaintainData

Create a temp file and add the line and save the file as a text file with name Lens.mf

Main-Class: MaintainData.LensView

To create a JAR file issue the command

jar cmf Lens.mf LensView.jar

**COMPILING THE PROGRAMS**

**Continuation Page ………**

Now run the program by executing a command line as

java –jar /path/to/file/LensView.jar

The jar file can also be runby just double clicking on the desktop.

The jar command is extensive and you will need to lookup some technical reference for several other options.

You will notice some class files are generated with $ in the name. Don’t bother these are anonymous classes. Any further problems locate an expert bugger at your locations.

**LOGFILES**: The LensViewApplication will initially create LogFiles under LiveDB.USRDIR your cwd. Later it

Will switch to LiveDB.serializedfilespath (top level directory you chose). In case of AUTOINSTALL look for

Top level directory LensView. The application will keep you informed of the log file name you must look for on disk. The Log Files will be very detailed for every error and will start with "LensViewLog\*.\*".

**SOURCE TWEAKS**

If you want to add more languages or delete some then you open InstallDB.java and search for method.

protected static void selectLanguages ()

Now follow all the chained methods and associated methods in the documentation of this method.

If for some reason there are problems with mailboxes (mails not sent) then open LiveDB.java and search for method protected static boolean PrepareAndSendMailMesg

In the above method one of the parameters is String p\_frompasswd which is set in the same method down below as password = GMAILPASSWORD; This GMAILPASSWORD is set at the top of LiveDB.java as

protected static String GMAILPASSWORD = "acrabacadra506"; Change this to your password and compile and create the jar files.

You might want to change the name of this application from LensView to some name you fancy. So to do that open VersionControl.java and right at top of the file search for ApplicationName = "LensView";

Found it okay now replace LensView with the name you fancy. Don’t make it too long though.

Okay you might face some problems with file locations then get some expert bugger to take a look at LiveDB.java method protected static String convertFileName

Most of the glitches I talked about earlier in this document should not create problems really. But never know while testing ?? For major changes to source code I will release another document later if time permits after Beta-Testing.

**SOURCE TWEAKS**

**Continuation Page ………**

There are adequate Bells and Whistles in the LensView Application including Sabotage and Intruder Alerts. But you might wish to have a fancier version of the Error Logging Method in LiveDB.java. This

Method is called grasshopper. Search for public static String grassHopper (Object... ex\_Exception).

This method also doubles up as a rudimentary debugger for even those who are not very conversant with usage of debugging tools.

Okay you might want to define some more properties in the LENSVIEWPROPERTIESFILE.txt and get the

Application to pick those up.

Then open InstallDB.java and search for protected static void getAppProperties()

Also search for protected static void readconffile () in InstallDB.java

Also search for protected static void writeConfFile () in InstallDB.java

The above three methods will need modifications if you choose to define more key=value pairs in your

LENSVIEWPROPERTIESFILE.txt. For any more key=value pairs define them at the top of class LiveDB.java

As protected static String IDENTIFIERNAME = null;

Animations are sometimes very complex and there is no way to understand the speed at which to draw each of the animation frames. All I have done is dissect the animation file and draw the frames one at a time in sequence. I am not looping the animations for now in the code. You might want some expert bugger to look up the source code for the GUI starting with FirstPage.java and do the looping for you.

Search for protected static boolean PlayAnimation (String p\_size, String p\_path, JPanel p\_panel)

Similarly I have not looped the audio player. Open FirstPage.java and search for

protected static boolean PlayAudio (String p\_path, JPanel p\_panel, String p\_size)

Don’t overload the movie files with long movies short advertisements should suffice. Gradually take it to a full length movie after initial testing is completed. Open FirstPage.java and search for

protected static boolean PlayVideo (String p\_size, String p\_path, JPanel p\_panel)

If you make changes to supported currencies then open CurrencyPanel.java, InvoicePage.java, SearchResultsPage.java, CataloguePage.java, SaleItemsPage.java and FirstPage.java and search for method protected static float convertCurrency(float p\_amt)

For any further source tweaks contact some expert bugger in the Bay Area.

FINALLY A THANK YOU TO ALL THE MUSICIANS WHO PLAY MUSIC TO MY EARS AND MADE THE IMPOSSIBLE POSSIBLE !!

I DEDICATE THIS DOCUMENT TO MY TWO DOGS SKIP (MALE) AND BELLA (FEMALE). SKIP IS A Phd IN STEALING LADIES HANDBAGS AND BELLA WARNS ME OF NASTY FEMALES AT THE DOOR. SKIP IS A GOLDEN RETRIEVER AND BELLA IS BLACK AMERICAN COCKER SPANIEL. SKIP IS MY HEARTBEAT !!!