

PBXLicense

Version 1.0

User Manual

Copyright © 2023 www.PAHLabs.com

Table of Contents

About	3
Support	4
Order	5
Getting Started	6
Main Screen	9
System Menus	10
Product Screen	11
Client Screen	13
View License	15
HTTP To Website	16
Configuration Screen	17
SMTP	18
SMTPDetail	19
Themes	20
Theme Detail	21
Background Textures	23
About Screen	24
License Deployment	27
Database Requirements	29
Code Inclusion	30
Screen Changes	31
Index	33

About

PBXLicense

PBXLicense is a **Complete Turn Key Solution** PureBasic customer licensing schema. I provide you cop to insert into your software application, which created the Client Demo License on first execution, and the About Box Licensing Form and Client License input screen. This system handles the License Encryption/Creation of Unique Product Keys, and Client Key Generation, as well as the launching of the keys to Clients Via Email.

This system tracks the many to many Products to Client Relationships. Select a Product and you can easily view the clients who own it. Select a Client and you can easily view what products they own.

Support

Support is through the website www.PAHLabs.com

and through the Purebasic Forum, and you can reach me through email at Pete@PAHLabs.com.

Order

Ordering will soon be available via the website. www.PAHLabs.com

Getting Started

Getting Started ... Quick Start!

The accompanying Code and DLL are located in your {INSTALL Directory}\Implementation

Step 1: Product

Create your Product.

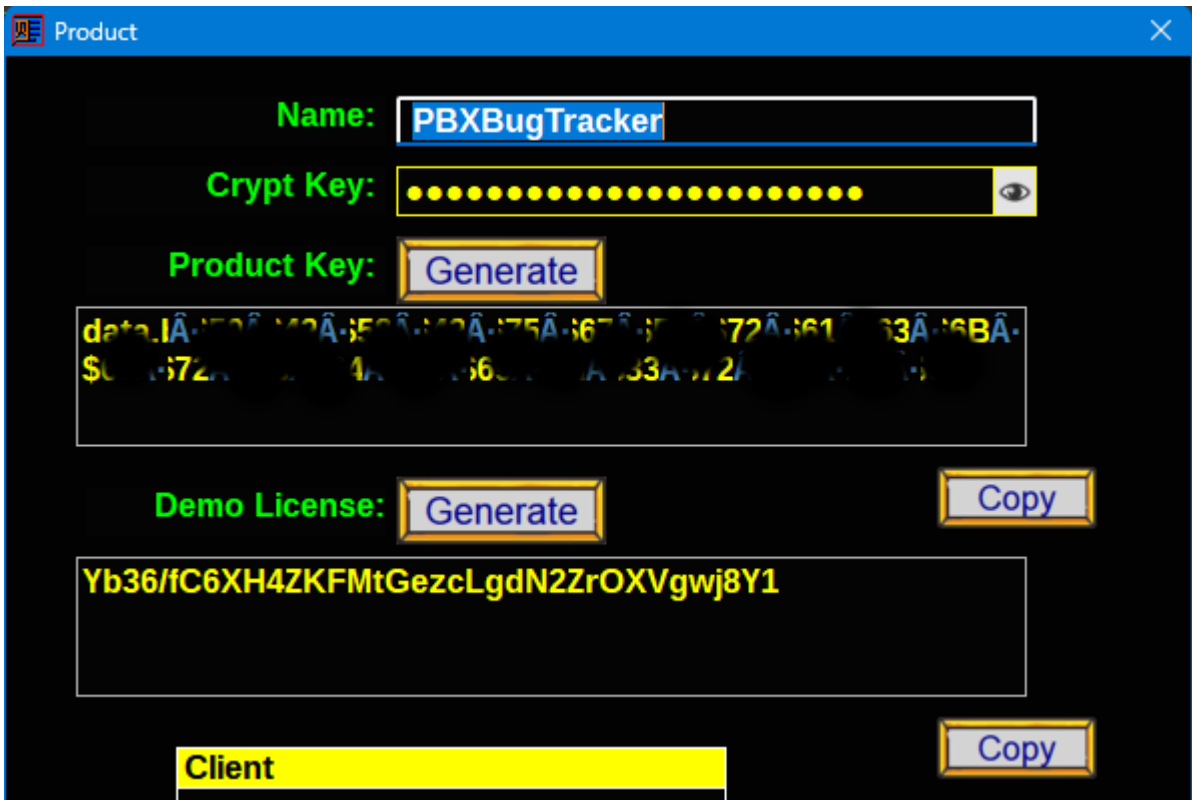
It needs a Unique Name, and a "Good" License Key.

By Good I'm thinking a mix of Alpha (Upper and Lower Case), Numeric, Symbols and be 15-25 Characters in length.

Once that is completed Click on the two Generate Buttons.

One generates your "Demo" License Key,

The other Generates your password for the code as a DataSection list of binary values.



I've destroyed the binary in the above screen shot.

The "ProductKey" (Binary data Section) will need to be Copied into the Data section of the ProdLicense.pbi" Script.

Step 2: Database Storage

There are three required fields in your database that this System Uses:

RegName - Name of Client

RegKey - Client encrypted license key.

RegImage - the cute Image for your About Box.

I store all of these on the Configuration Table. You can store them anywhere you like. Just modify the code in ProdLicense.pbi Accordingly.

Step 3: Code Update to your Product.

You will need to include 4 Files into your Project., However only 1 Additional file is included in your deployment.

Cipher.dll - The Encryption Cipher DLL. Used in Compile, and **MUST** be Deployed with Product.

Cipher.lib - Used in Compile ONLY.

chi_Cipher.imp - Used in Compile ONLY.

ProdLicense.pbi - all Structures, Procedures to handle product Licensing.

Additionally there are the Additions to the About Box, and License Entry Screen

Step 4 - Assign Licenses to your Clients

As people/Clients purchase Licenses, you enter in their Name and Email Address, Then simply Check off what products they own, Upon Saving the Client Record, the Licenses will be generated. Select Client, and License and Email Away. Email requires con figuration, see Configuration section on Configuring your Email SMTP.

Client

PBXBugTracker

First:

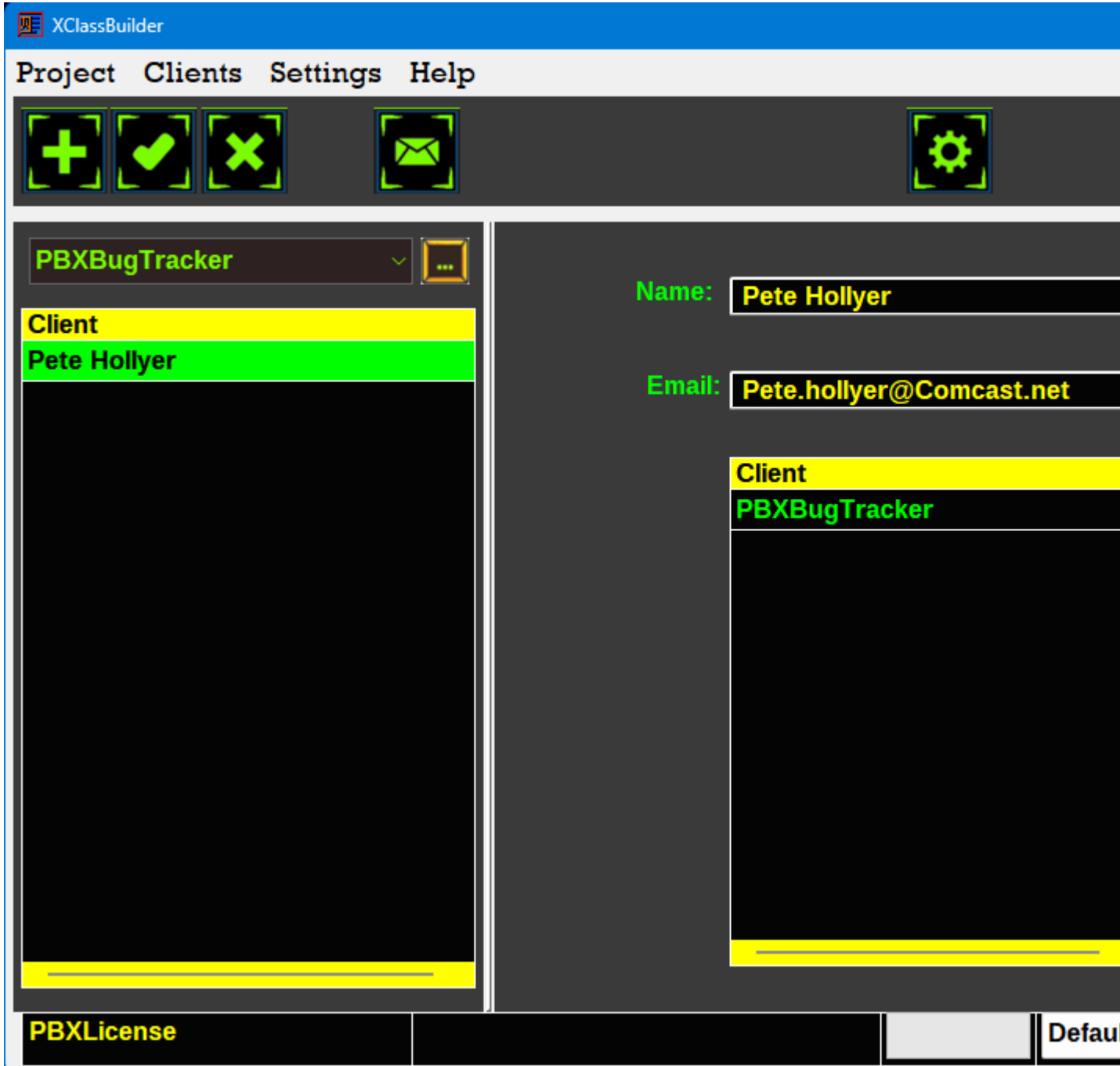
Email:

Owned Products

	Name
<input checked="" type="checkbox"/>	PBXBugTracker
<input type="checkbox"/>	PBXClassBuilder
<input type="checkbox"/>	PBXLicense

Main Screen

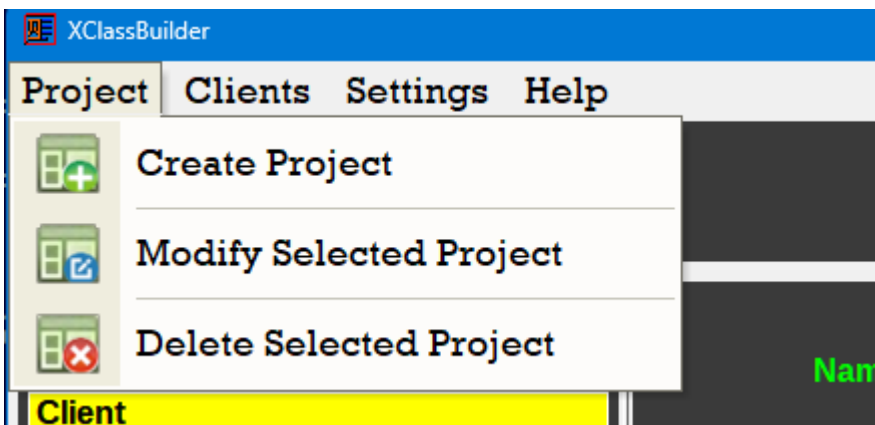
Main Screen



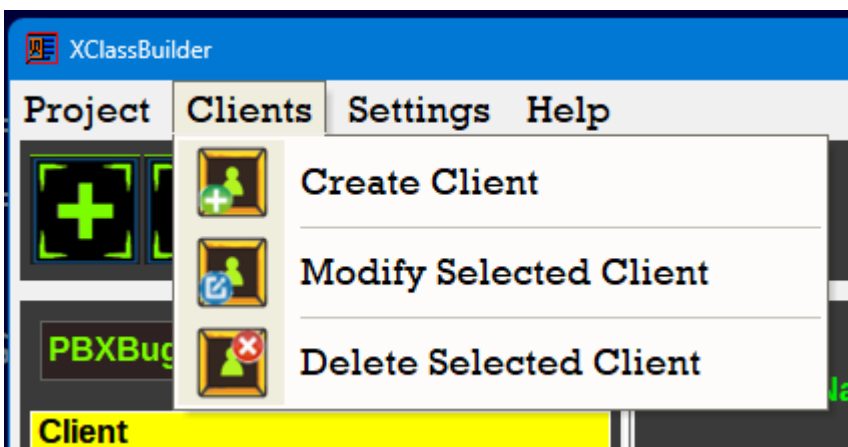
System Menus

System Menus

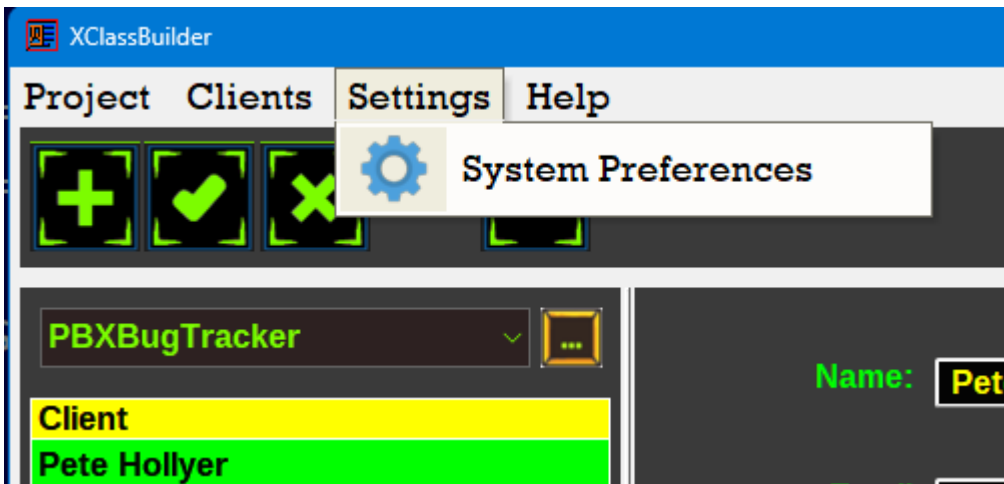
Project Menus



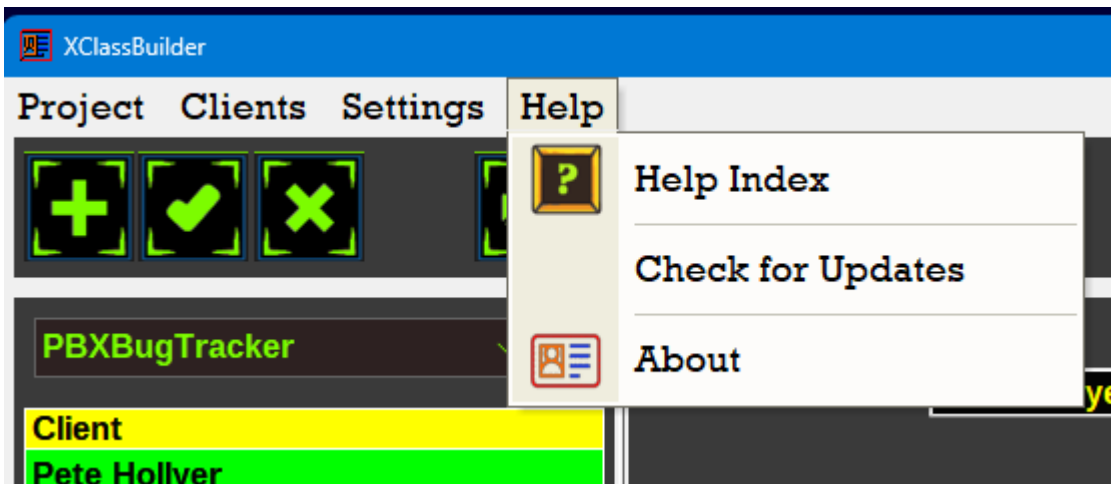
Client Menus



Preference Menu



Help Menus



Product Screen

Product Screen

Each Product needs a Unique Name and Crypt Key. Crypt Key should be 15-25 Characters in Length, Upper and Lower Case, Numbers and Symbols.

The screenshot shows a software dialog box titled "Product". It features a blue header bar with a close button. The main area is black with green text labels for "Name:", "Crypt Key:", "Product Key:", and "Demo License:". Each label is followed by a corresponding input field or button. "Name:" has a text box. "Crypt Key:" has a text box with a visibility icon. "Product Key:" has a "Generate" button and a large empty text box. "Demo License:" has a "Generate" button, a "Copy" button, and a large empty text box. A "Client" list box is highlighted in yellow, containing a single empty entry. At the bottom are "Save" and "Cancel" buttons.

Product Name - Unique Product Name.

Crypt Key - Unique Key for THIS Product to Generate Client Licenses. It needs a Unique Name, and a "Good" License Key.

By Good I'm thinking a mix of Alpha (Upper and Lower Case), Numeric, Symbols and be 15-25 Characters in length.

Generate Prod Key - Since Strings are visible inside the binary exe, This system does NOT use a String for these keys. Instead is converted into Binary and put into the program as a Data Section Load, where it is eventually converted back into a string for usage.

Generate Demo License - On FIRST Execution of the Check License, if Empty the system will Generate a "Demo" License key, In code you can set the duration of time allotted for the License. The Field here can be used to visually inspect, for validation in testing.

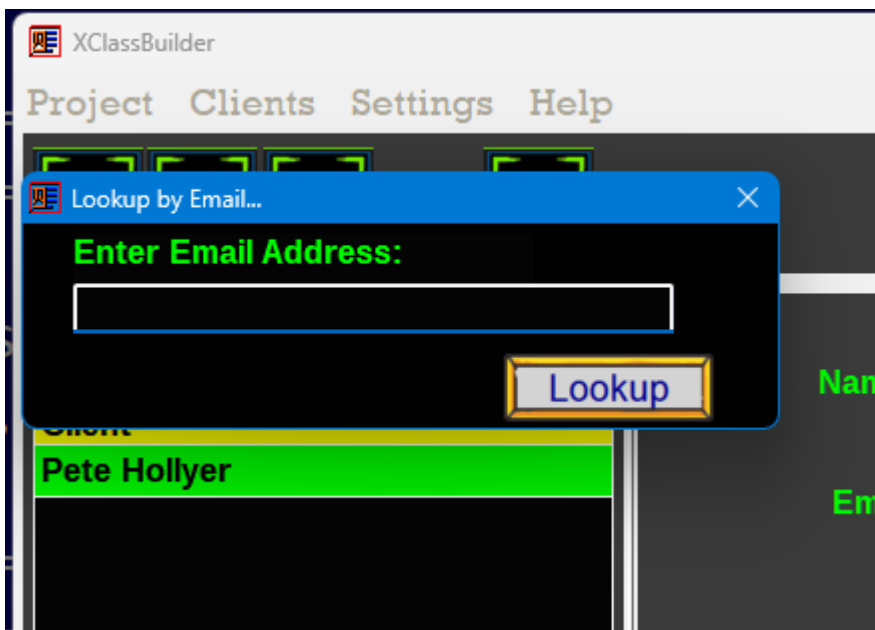
Finally - List at bottom of form are the clients wov you have created Licenses for this Product.

Client Screen

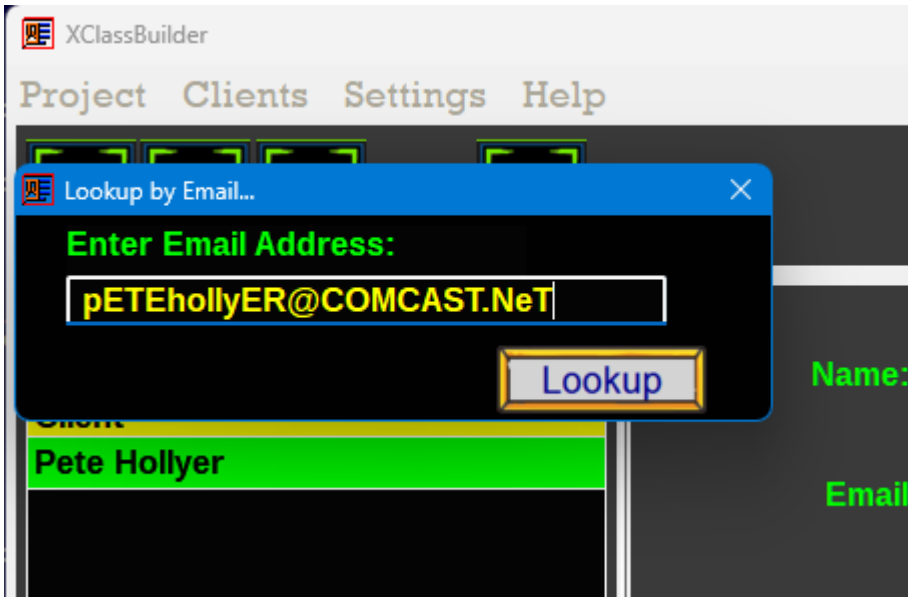
Client Screen

Clicking ADD or CREATE, will ask you to enter the Email Address your wanting to set up. **This is a Case Insensitive entry!**

If the address exists under som other Product, then that Record will be retrieved to Update, otherwise the system will simply CREATE a Client as Expected.



Case Insensitive Email Address entry



Still retrieves:

Client

BugTracker

First:

Email:

Owned Products

	Name
<input checked="" type="checkbox"/>	BugTracker
<input checked="" type="checkbox"/>	ClassBuilder
<input checked="" type="checkbox"/>	PBXLicense
<input checked="" type="checkbox"/>	PBXSpriteSheet

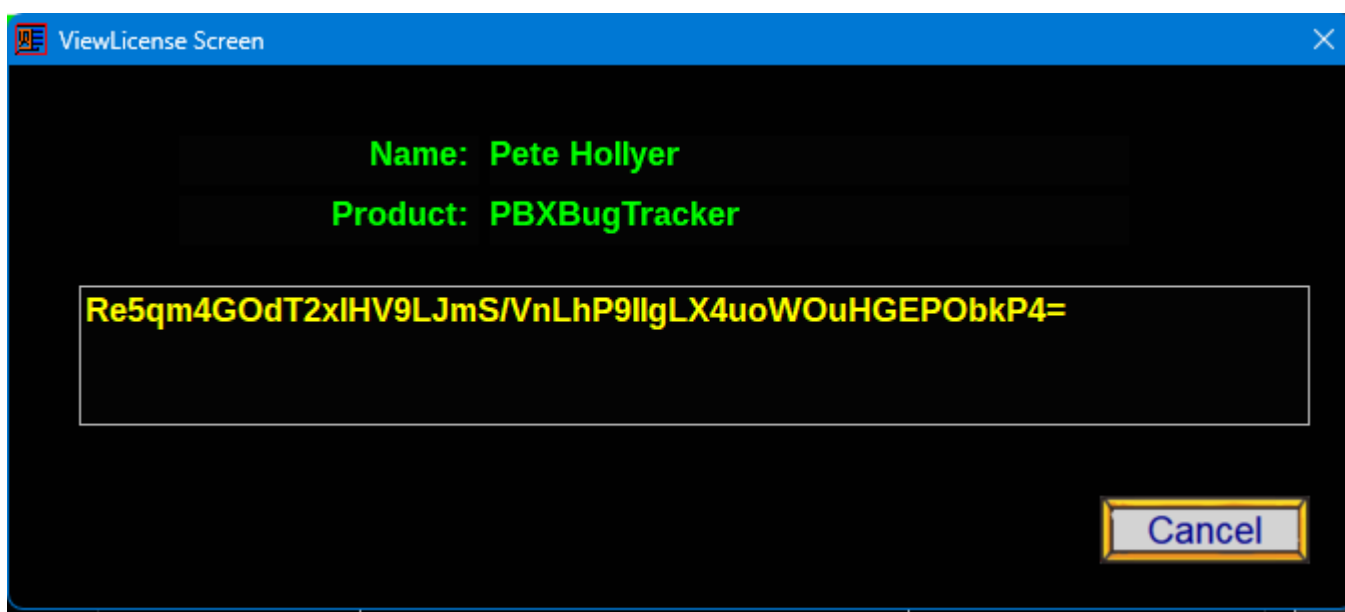
Note: Once a key has been created the Person's Name is LOCKED. This is required for key validation inside your application.

Use Check Boxes to activate License Creation ON SAVE. Once Saved, you can Email out Licenses to Client/Customer.

View License

View License

Clicking on the VIEW button will bring you the following four your Review



HTTP To Website

HTTP To Website

A Version 1.2 Enhancement. This allows a HTTP GET to be sent to your website to update your client's personal records.

It is customary that when a person purchases a Product from you that they can lookup their licenses from their account on your website. To facilitate this I have added this feature to this Product.

There are accompanying changes to the system configuration screen. These changes let you configure a rest API on your website, these include URL, Authorization Code, and if you want results displayer in popup message.

The HTTP Payload is a standard JSON file.

```
{ "ProductName":"productname", "ClientName":"clientname",
  "ClientEmail":"emailaddress", "LicenseKey":"clientLicense"}
```

Once your website Rest API receives get request it should confirm Client Email is correct, then update a appropriate Table with the License Data.

Response JSON Should be in the form:

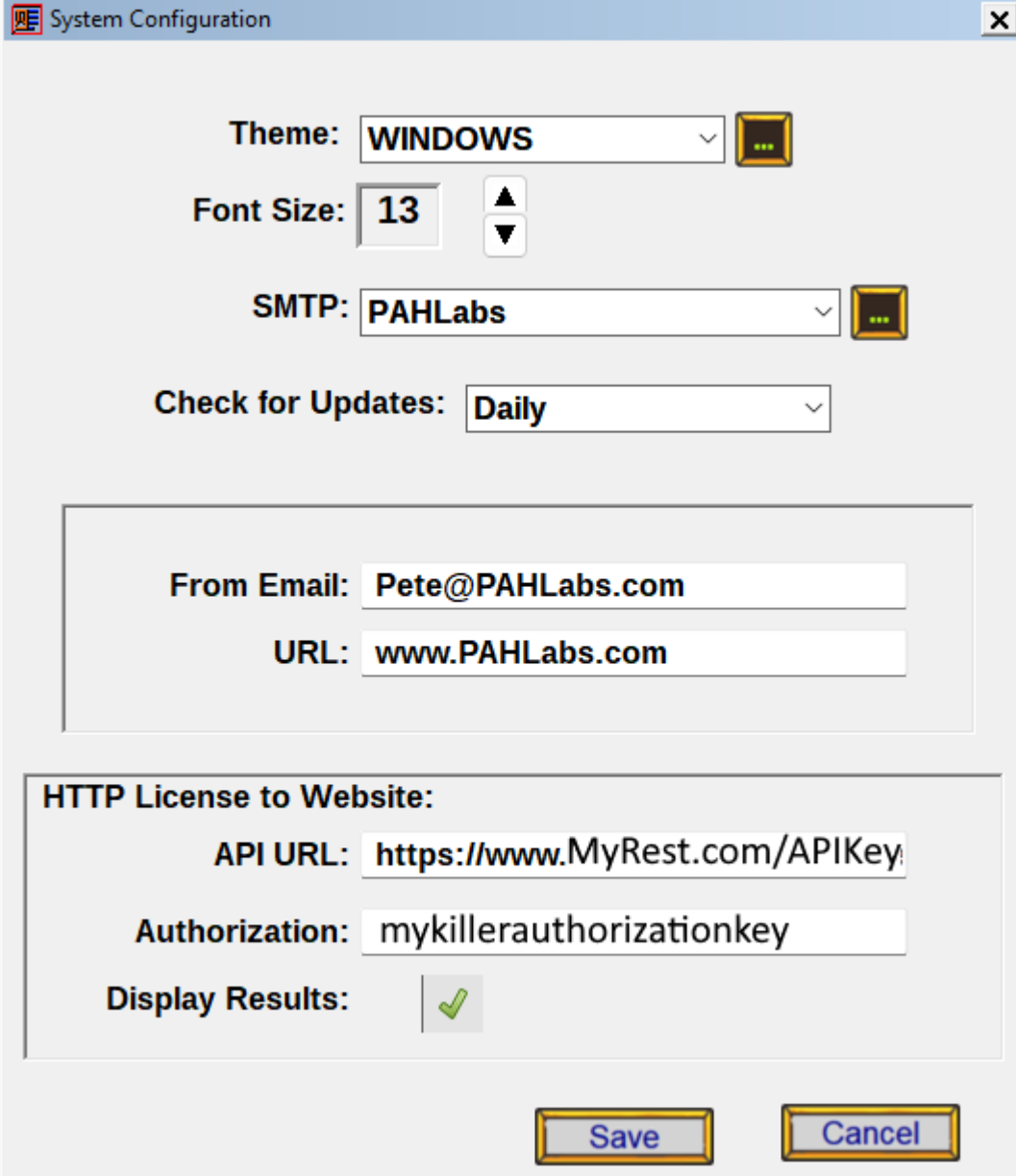

```
"status": "success", "data": "Appended"}
```

Values are up to you.

Configuration Screen

Configuration Screen

System Configuration screen



The screenshot shows a 'System Configuration' dialog box with the following fields and controls:

- Theme:** A dropdown menu set to 'WINDOWS' with an ellipsis button to its right.
- Font Size:** A text input field containing '13' and a vertical spinner control.
- SMTP:** A dropdown menu set to 'PAHLabs' with an ellipsis button to its right.
- Check for Updates:** A dropdown menu set to 'Daily'.
- From Email:** A text input field containing 'Pete@PAHLabs.com'.
- URL:** A text input field containing 'www.PAHLabs.com'.
- HTTP License to Website:** A section containing:
 - API URL:** A text input field containing 'https://www.MyRest.com/APIKey'.
 - Authorization:** A text input field containing 'mykillerauthorizationkey'.
 - Display Results:** A checkbox that is checked, indicated by a green checkmark icon.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

Theme - Select your desired theme. Use ellipse to edit/create themes

Font Size - System wide Font size used.

SMTP Selection - this will be the selected SMTP for emails. Use Ellipse to edit SMTP Gateways

Checking for Updates - how frequently you desire for the system to check for updates.

Email From - The Email address in the "From" Address field

Your URL - this will be Your website URL, this is part of the email body.

Version 1.2 HTTP Enhancement

For use with website updating I've added an HTTP Get to the System

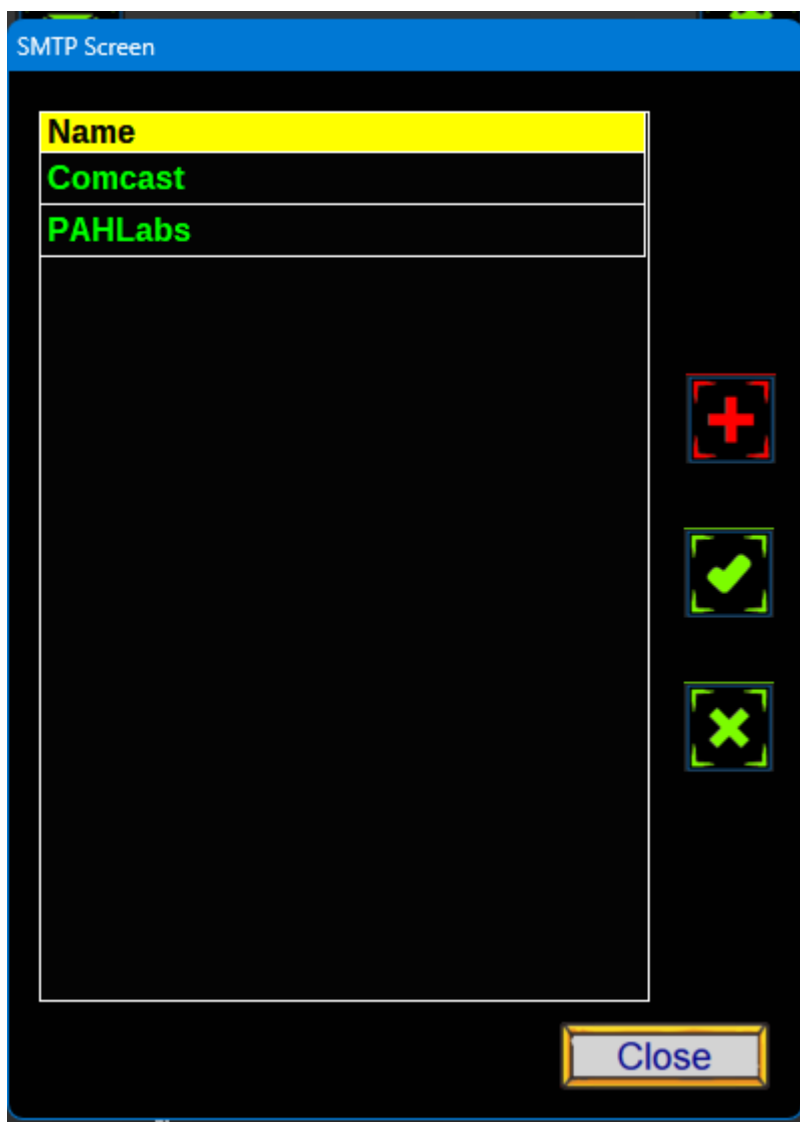
API URL - Enter in the website Rest API.

Authorization - Enter in the API Rest Authorization key

Display Results - Allows HTTP Get Response to be displayed as a Popup message.

SMTP

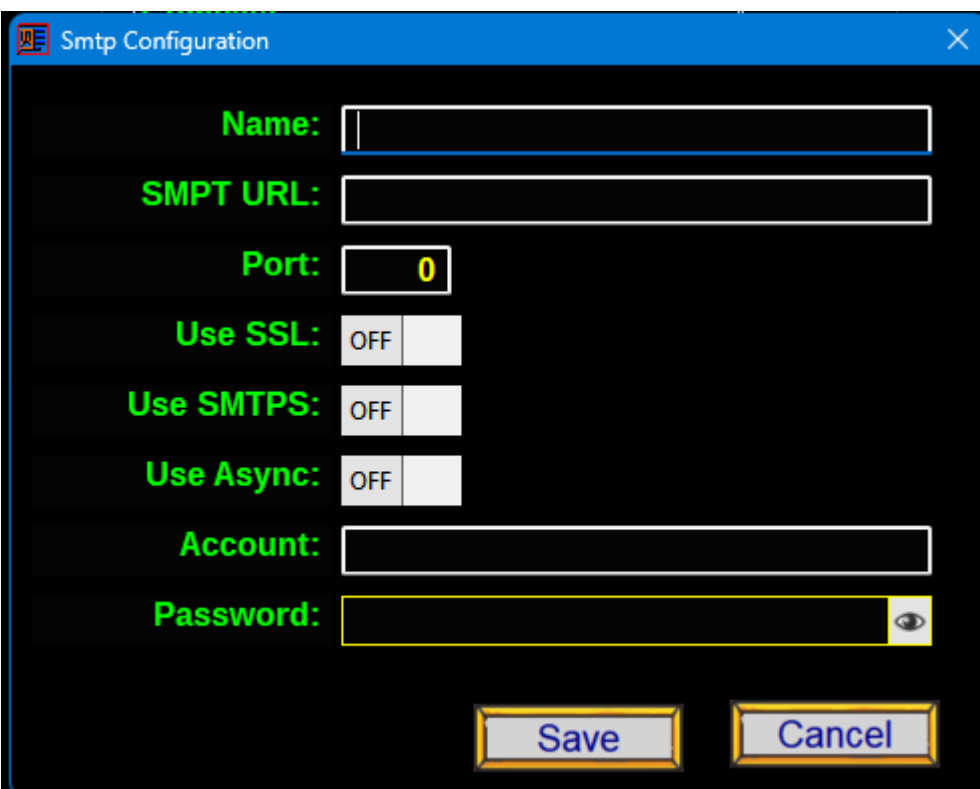
SMTP



It IS possible to have more than One SMTP Gateways. It's up to you how many you setup. However the "Default" USED will be the selection on the Configuration Screen.

SMTPDetail

SMTP Detail



The image shows a dialog box titled "Smtplib Configuration" with a close button (X) in the top right corner. The dialog has a black background with green text for labels. The fields are as follows:

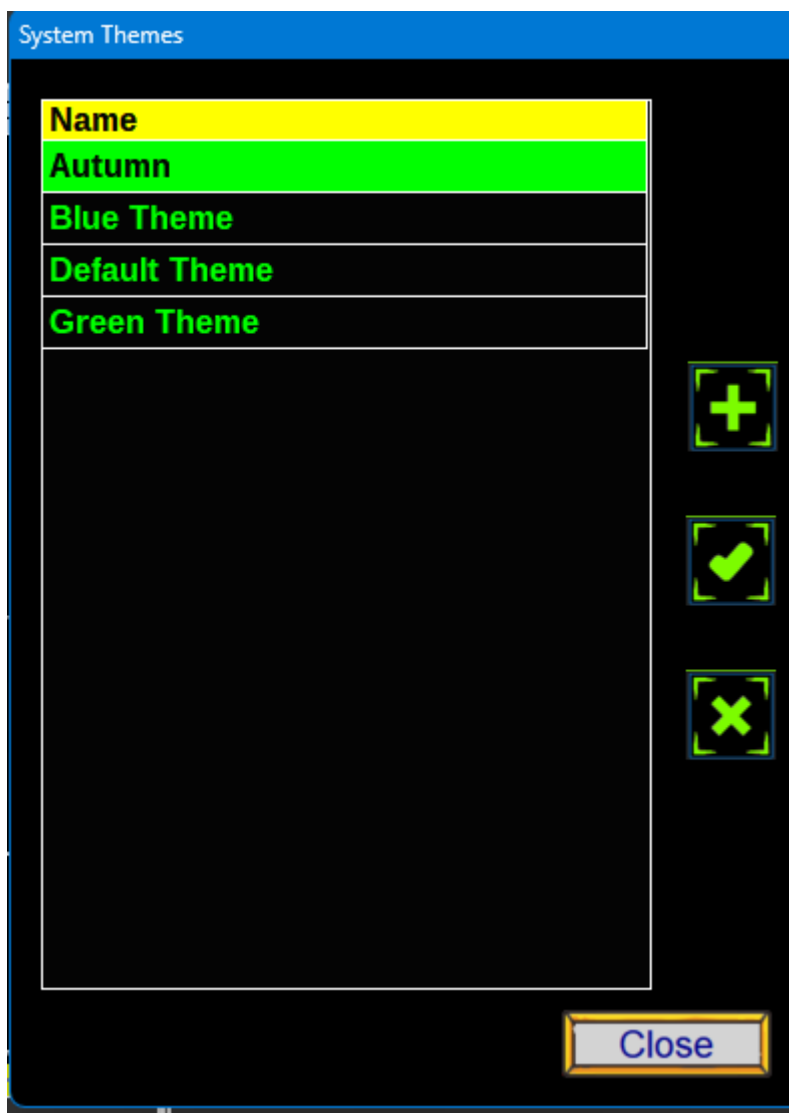
- Name:** A text input field.
- SMTP URL:** A text input field.
- Port:** A numeric input field containing the value "0".
- Use SSL:** A checkbox labeled "OFF".
- Use SMTPS:** A checkbox labeled "OFF".
- Use Async:** A checkbox labeled "OFF".
- Account:** A text input field.
- Password:** A text input field with a visibility toggle icon (an eye) on the right side.

At the bottom of the dialog, there are two buttons: "Save" and "Cancel".

You need to google your email SMTP, but I'll suspect you already know this information as it is part of all websites.

Themes

Themes



This is the standard frmList Screen, I'm debating adding a Clone field so that you don't have to event the wheel when creating Themes.

Theme Detail

Theme Detail

System Theme

Name:

Color or Texture

Form Back: Green 1

Container Back:

Text Back:

Text Fore:

String Back:

String Fore:

Combo Back:

Combo Fore:

Editor Back:

Editor Fore:

Editor SpellCheck:

Editor Selection:

List Back:

List Fore:

List Lines:

Header Back:

Header Fore:

Row Back:

Row Fore:

Row Lines:

Scroll Back:

Scroll Fore:

Here you choose the colors you desire for the many selections. Once created the theme will be selectable and useable.

Background Textures

Background Textures

Clicking the ellipse button aside the Background textures selection of the theme Detail screen will bring up the Background Textures List Screen



Use the buttons on right to manage the list.



Background textures are 200 x 200 image, used to tile screens. must have Unique Name.

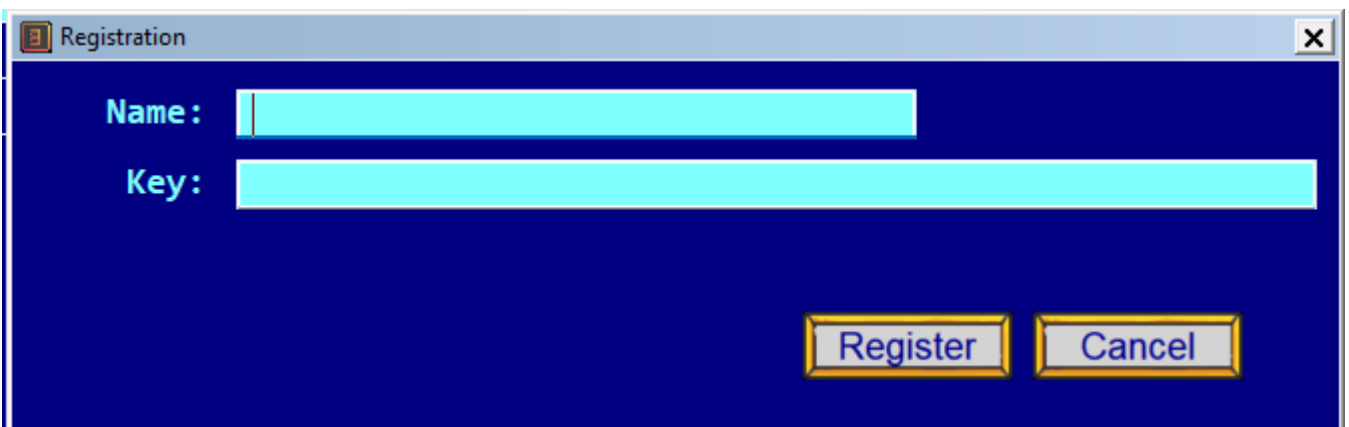
About Screen

About Screen



Here you are informed about the current Code Version you are running.

Also you can Review your License., Initially you will have a Guest License, Clicking on the "Registration" Button will bring up the Registration Screen.,



Enter Your Name **EXACTLY** as it appears in your Registration Email.
Enter your License Key, I **Strongly** recommend using **Copy** and **Past**.

When entered, your name and License will appear in the Image.

License Deployment

License Deployment

The accompanying Code and DLL are located in your {INSTALL Directory}\Implementation

This is this most important of the sections in the entire help. If you read anything read this.

All the called code modules reside in a single Code Script Sheet. Namely "ProdLicense.pbi" which is short for Product License.

This script should be included prior to your main screen is Displayer. Where it resides in your project is not important, so long as you path it's include properly.

To include this script enter the following:

```
XIncludeFile "Views/frmExpire.pbi"  
XIncludeFile "ProdLicense.pbi"  
XIncludeFile "Views/frmAbout.pbi"
```

The Script ProdLicense.pbi is fully documented.

You MUST input your Product Key in the First DataSection, AND update the Character Count

```

;=====
;=====
;=====
;====      Enter Product License HERE!!!
;=====
;=====
;=====
;=====
;=====
DataSection
  PWDSize:
  Data.i 20
  PWD:
  Data.b $50, $40, $69,
EndDataSection

```

Character Count input

Product Key Bin

A "standard Create Check of License is This:

```

Expires=CheckLicense()
If Expires > 0
  THEMEDMESSAGE::MessageBox(GetAppName(),
                           "Demo will Expire on ")
EndIf

```

Expires is defined as a Long.
This includes the Nag screen.

On your About Screen, when Clients Click "Register", Here is the needed code:

```

Case #Gadget_frmRegistration_btSave
  Select EventType()
    Case #PB_EventType_LeftClick
      ;Confirm License Key
      If RegisterLicense(GetGadgetText(#Gadget_
                          GetGadgetText(#Gadget_
CONFIGURATION::Find(1, @SystemConfigura
HideGadget(#Gadget_frmAbout_btRegister,
;Update About Image
If IsImage(SystemConfiguration\RegImage
  SetGadgetState(#Gadget_frmAbout_picLi
  SetGadgetState(#Gadget_frmAbout_picLi
ImageID(SystemConfigura
  DisableWindow(#Window_frmAbout, #Fals
  CloseWindow(#Window_frmRegistration)
  SetActiveWindow(#Window_frmAbout)
    EndIf
  EndIf
EndSelect

```

Database Requirements

Database Requirements

As mentioned in the quick start, there are only 3 fields required by this implementation.

RegName - Name of Client

RegKey - Client encrypted license key.

RegImage - the cute Image for your About Box.

I store all of these on the Configuration Table. You can store them anywhere you like. Just modify the code in ProdLicense.pbi Accordingly.

Index	Name	Declared Type	Type
> 1	ID	INTEGER	INTEGER
9	RegName	VARCHAR(150)	VARCHAR
10	RegKey	VARCHAR(250)	VARCHAR
11	RegImage	BLOB	BLOB

If you make modifications to the Crypt Payload Structure the more space will be required.

```

;=====
;=== Crypt Structure
;=====
Structure CryptPayload
    Name.s
    Expire.i ; 0 = Never
EndStructure

```

Basically Data is stored in instance of Structure, then Converted into a Json String, then Encrypted. The Encrypted Text is Stored in Database Field RegKey.

RegName contains the Unencrypted Name from the CryptPayload Name Field. This field is used to verify Valid Encryption/Decryption.

RegImage is the "cute" image to display on your About Screen to tell the user about their License.

Code Inclusion

Code Inclusion

Or in other words what do you need where to BUILD, and What you Deploy with your Product. I feel the Quick Start did a good job on this:

You will need to include 4 Files into your Project., However, only 1 Additional file is included in your deployment.

Cipher.dll - The Encryption Cipher DLL. Used in Compile, and **MUST** be Deployed with Product.

Cipher.lib - Used in Compile ONLY.

chi_Cipher.imp - Used in Compile ONLY.

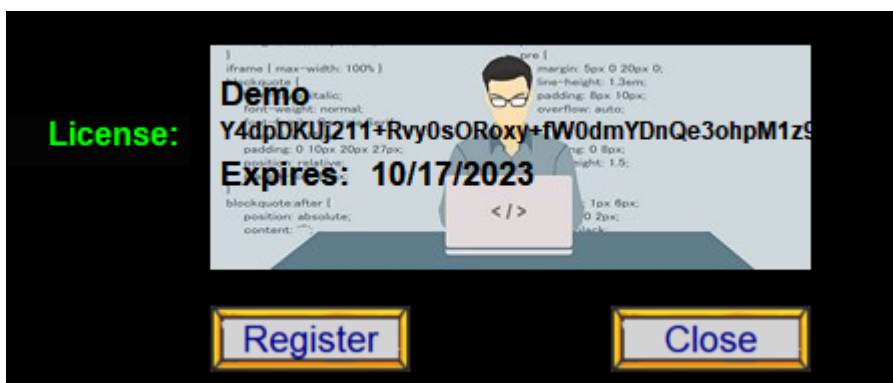
ProdLicense.pbi - all Structures, Procedures to handle product Licensing.

Additionally there are the Additions to the About Box, and License Entry Screen

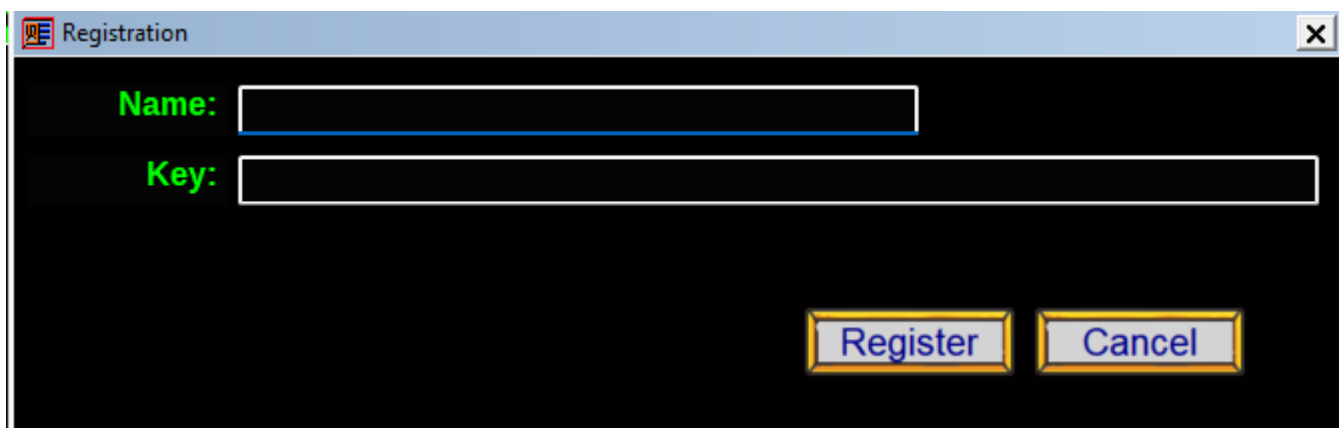
Screen Changes

Screen Changes

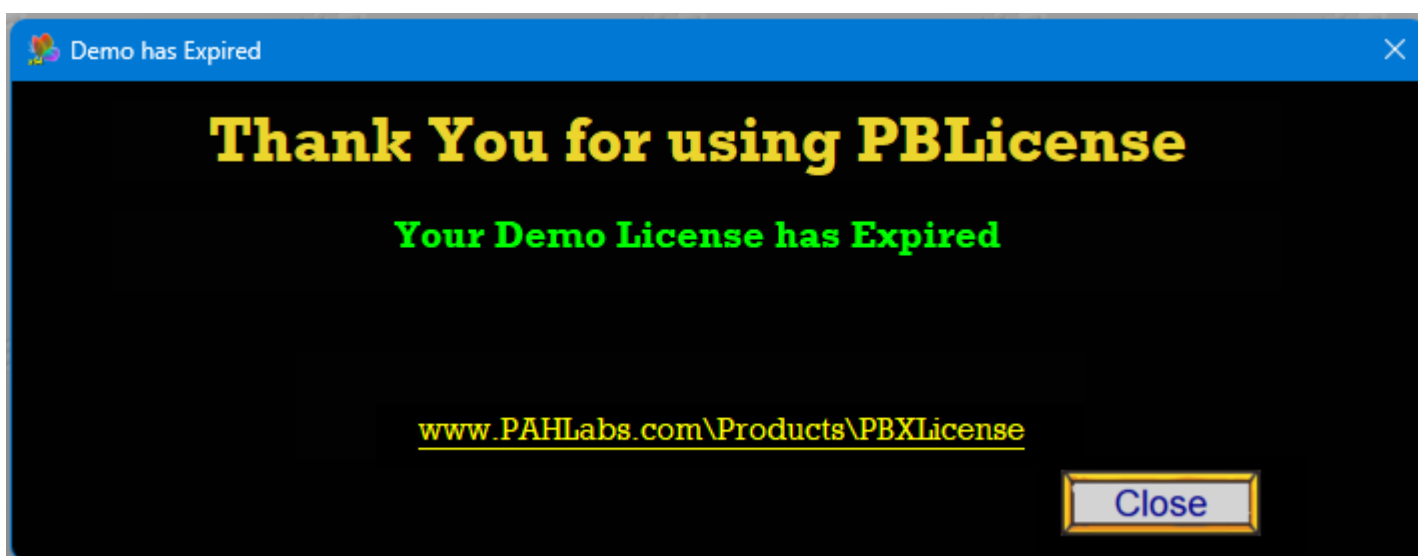
About Screen Changes:



Registration Screen



Expired Screen



Index

C

contact *4*

E

email *3, 4, 8, 13, 14, 15, 16, 18, 20, 25*

S

support *2, 4*

W

website *2, 4, 5, 16, 18, 20*