

# PIMPhony Integration Utility Guide

Prepared by Eko Wibowo

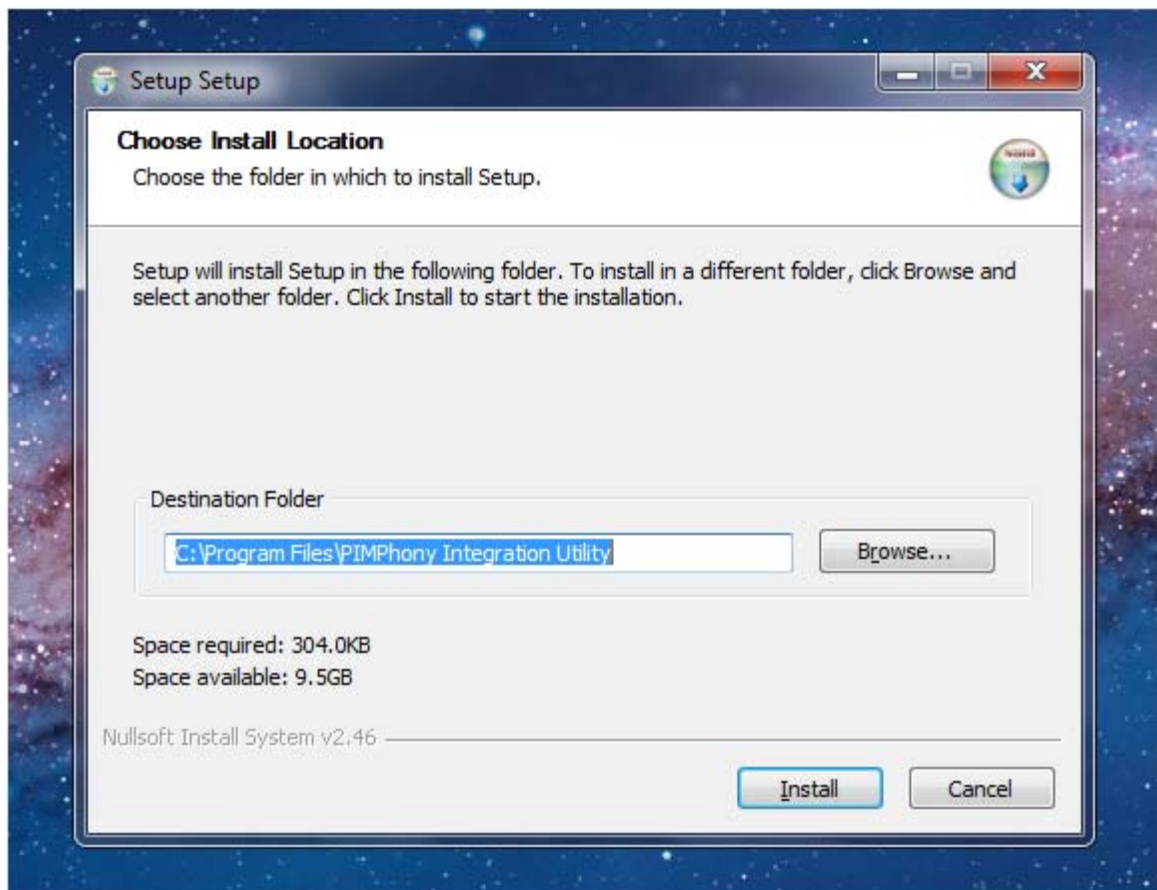
<http://bit.ly/swdev-cv>

# PIMPHONY INTEGRATION UTILITY GUIDE

This document will briefly guide you on integrating Pimphony with our-yet-to-develop integration utility. It will be revised based on job progress.

## INSTALLING PIMPHONY INTEGRATION UTILITY

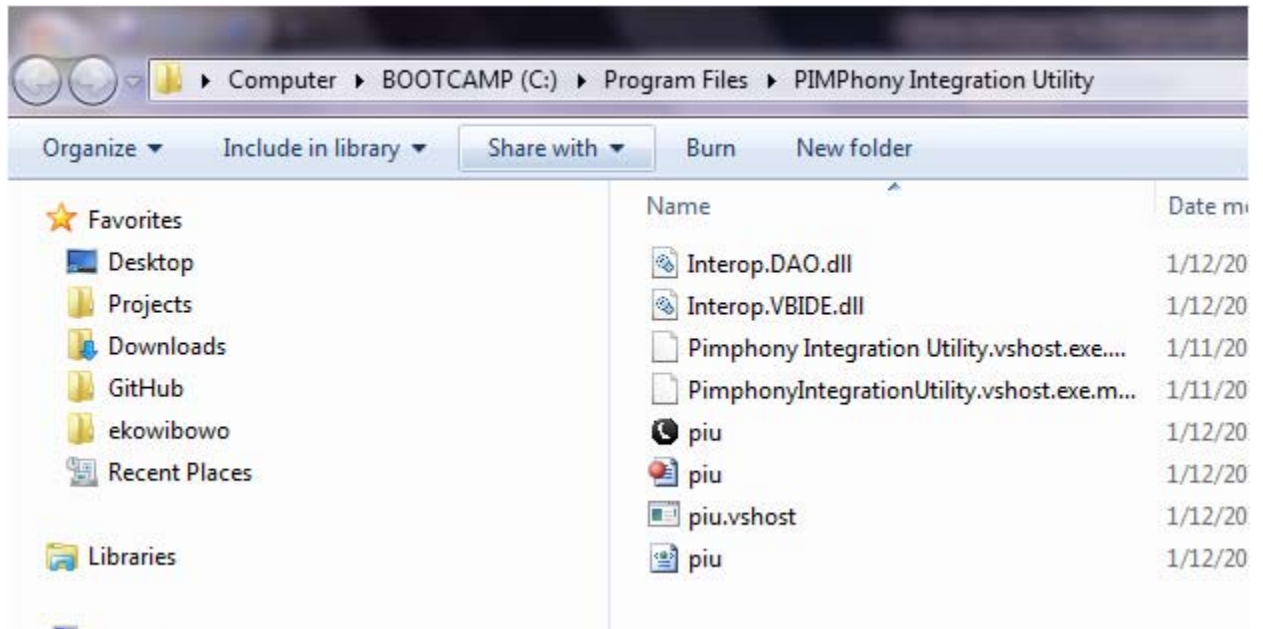
1. This application was built and developed using Windows 7. Hereby, it will run without problem if run on Windows 7. In older version of Windows, you may have to install DotNet Framework 3.5.
2. The application is distributed as single file Setup.exe. Just double click it, and the installation window will open as follows :



3. Before pressing install, please press CTRL+C (or right click on the highlighted text, and choose Copy) to copy the Destination Folder to the clipboard, you will paste it in the integration phase in PIMPhony.
4. Press Install to copy the utility into folder C:\Program Files\PIMPhony Integration Utility. And press Close in the next window.

## PIMPHONY INTEGRATION UTILITY GUIDE

5. If you open Windows Explorer to this folder, C:\Program Files\PIMPhony Integration Utility, you'll see the following content of the utility :



The file piu.exe (the one with white on black phone icon) is our executable file. You will put this information in the PIMPhony integration window.

6. Finish, now proceed to the integration phase

## CUSTOMIZING PIMPHONY TO CALL OUR INTEGRATION UTILITY

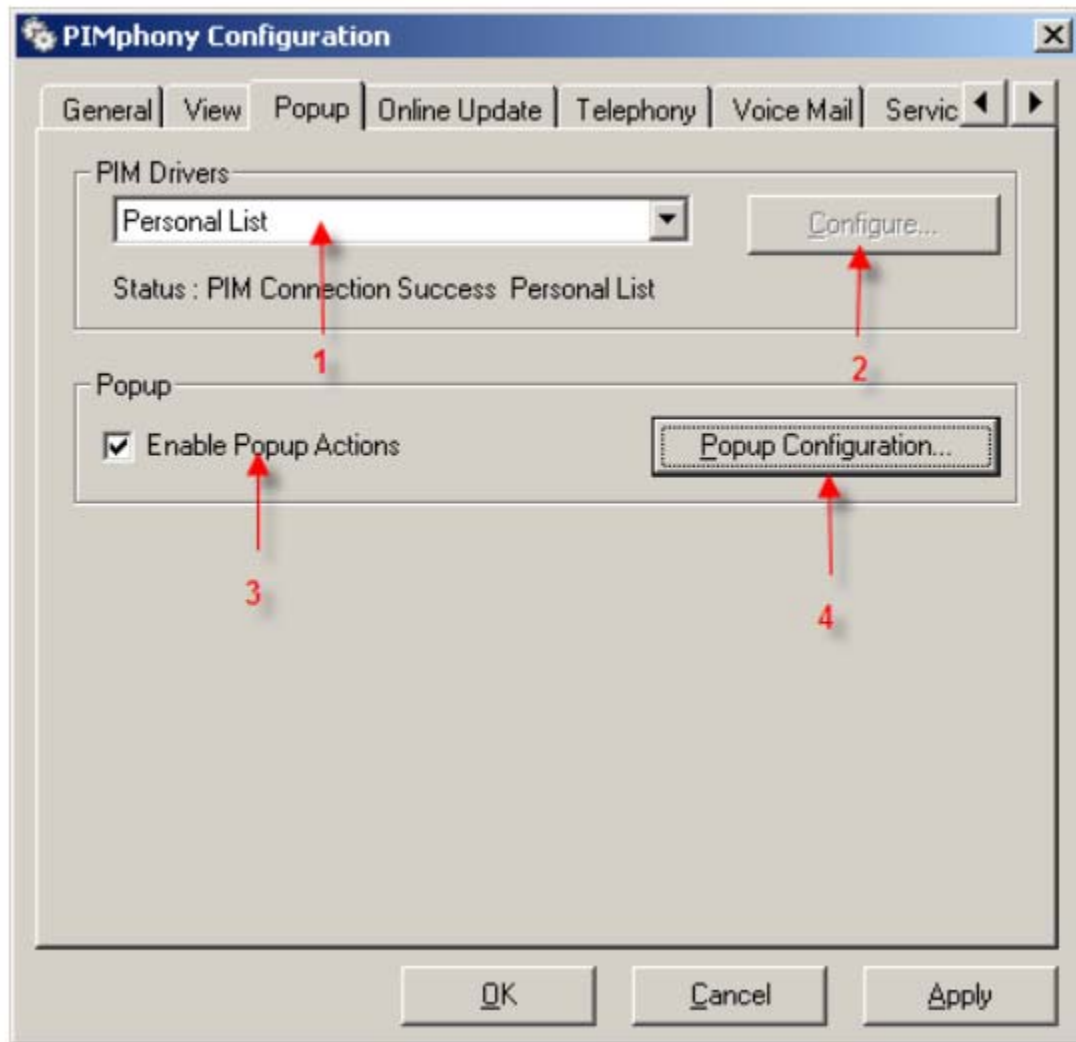
In creating this steps, I depend solely on the Alcatel's official PIMPhony Integration Developers Guide. I took some part of their screenshot there, and pasting it here.

Steps:

1. Access PIMphony configuration Window by choosing menu Tool -> Popup

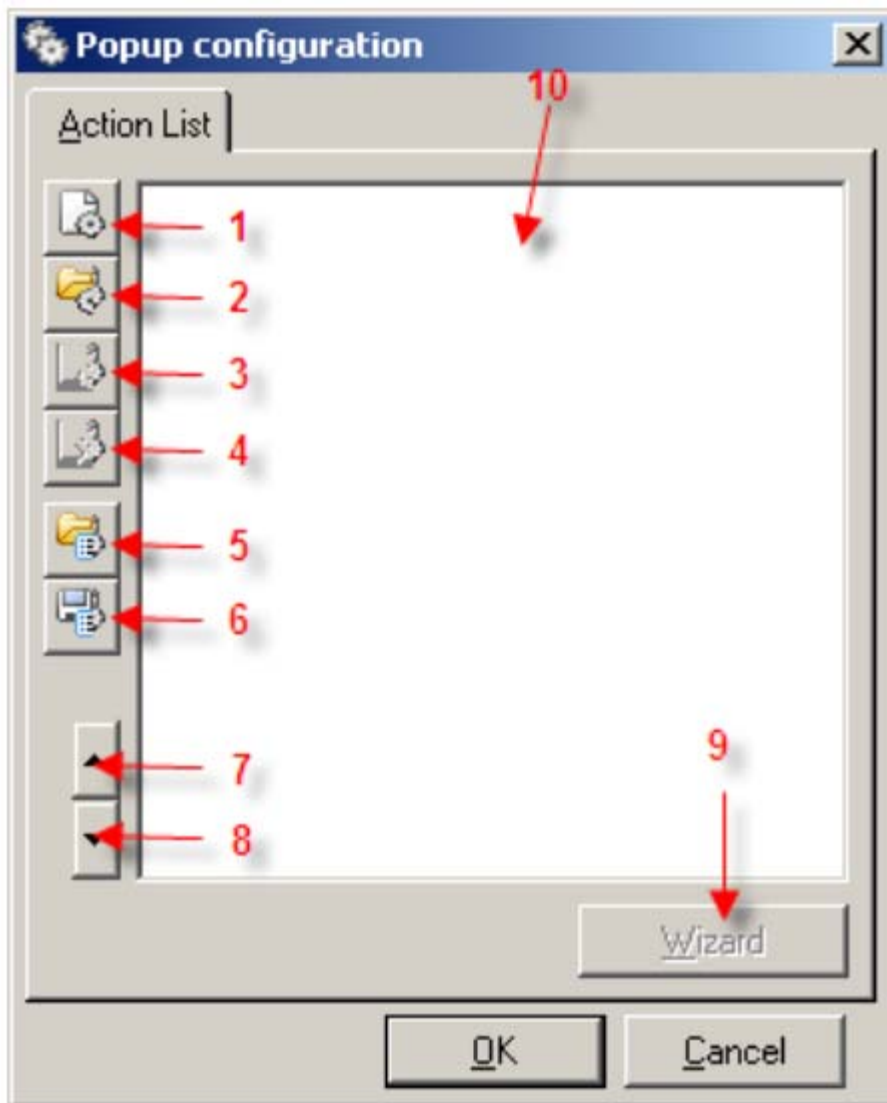
## PIMPHONY INTEGRATION UTILITY GUIDE

2. Click Enable Popup activations, and click Popup configuration



3. In Popup Configuration window that follows, klik Add New action button (numbered 1).

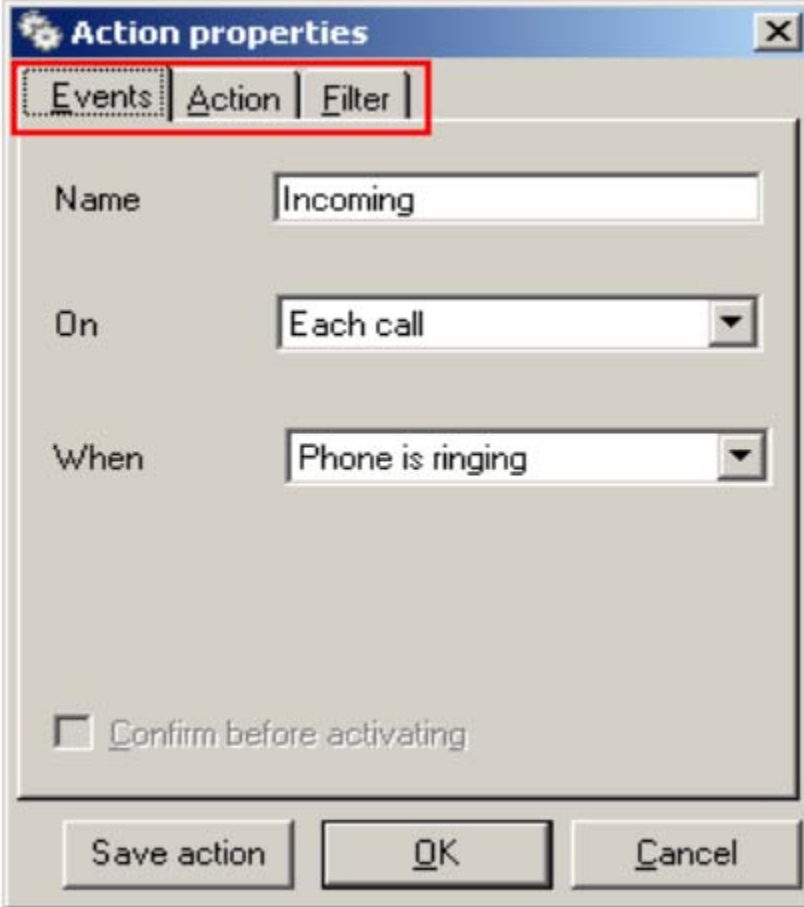
## PIMPHONY INTEGRATION UTILITY GUIDE



4. In the following Action Properties that follows, please setup the proper events that you want to handle. I predict what we need is On Each Call, and When Phone is Ringing. By testing, we will know the right configuration of

## PIMPHONY INTEGRATION UTILITY GUIDE

this event.



**Action properties**

Events | Action | Filter

Name: Incoming

On: Each call

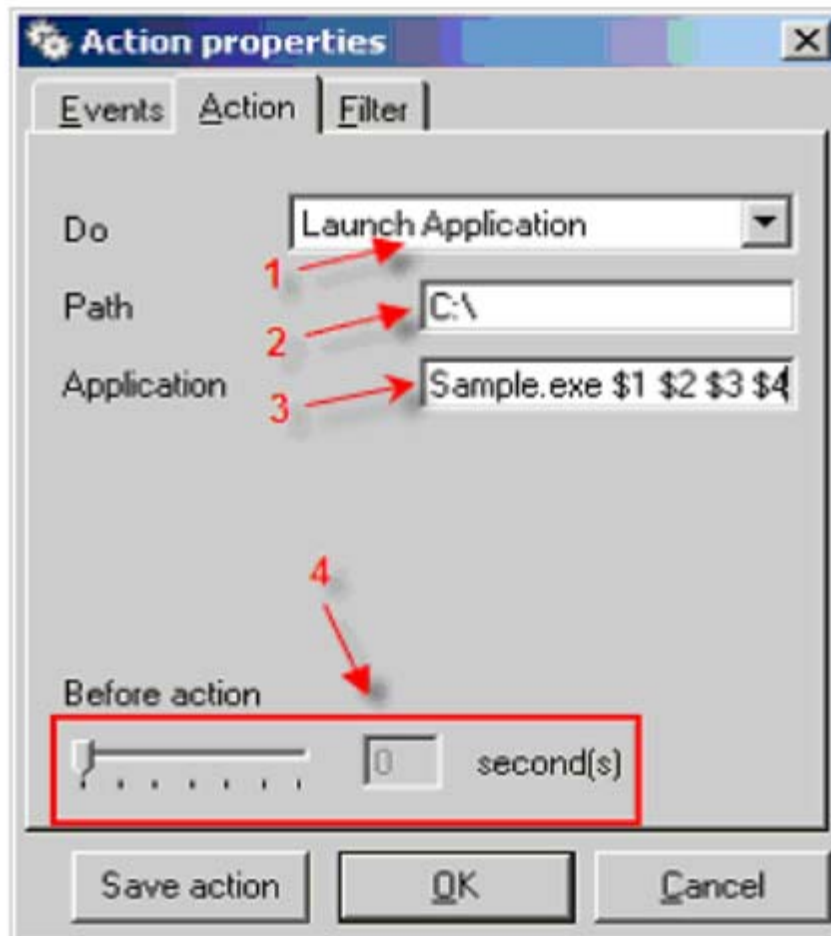
When: Phone is ringing

☐ Confirm before activating

Save action OK Cancel

## PIMPHONY INTEGRATION UTILITY GUIDE

5. This is the most important part : the Action tab.



Customized the action tab as shown above with modification in the Path textbox. Paste the path to our utility, **C:\Program Files\PIMPhony Integration Utility**, as copied in the last step of the installation of the utility.

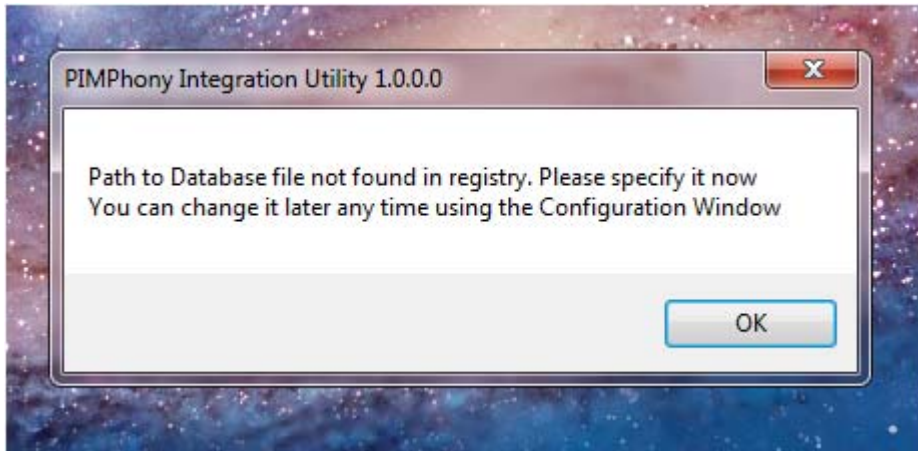
In the Application name, just type `piu.exe`

And, from this, I am not to sure. For now, please test it with this argument :  
`$1 $2 $3 $4`

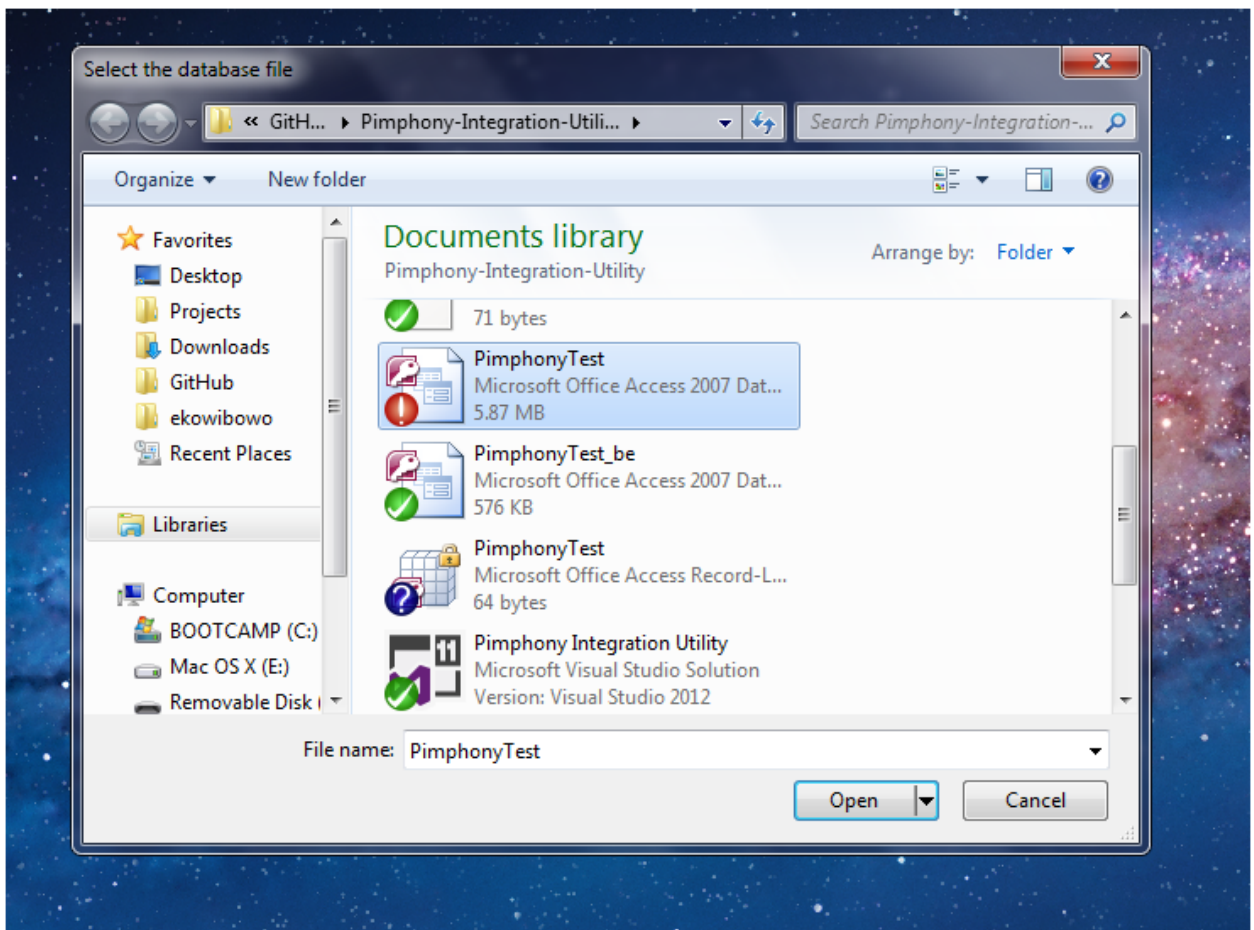
So the complete Application text box will be filled with :  
**`piu.exe $1 $2 $3 $4`**

## PIMPHONY INTEGRATION UTILITY GUIDE

- Now Press OK, and close the Action Properties window and also PIMPhony Configuration window.
- Now, you can test call PIMPhony. The first time a call is answered, you will see the following window :



Press OK, and browse for the PIMPhonyTest.accdb (or any database name that you want actually) :





## PIMPHONY INTEGRATION UTILITY GUIDE

Once, the correct database is selected, press Open.

- Now you'll be faced with the database window opened to the NewCall form with the caller number in the **Number called in on ..** field

The screenshot displays the Microsoft Access 2007 interface. The title bar indicates the file is 'PimphonyTest : Database (Access 2007) - Microsoft Access'. The 'Database Tools' ribbon is selected, showing options like 'Run Macro', 'Visual Basic', 'Relationships', 'Property Sheet', 'Object Dependencies', 'Database Documenter', 'Analyze Performance', 'Analyze Table', 'SQL Server', 'Access Database', and 'Link'. The 'All Access Objects' task pane on the left lists various database components. The 'NewCall' form is open, featuring a header with the title 'NEW CALL' and the date 'Sunday, January 13, 2013'. The form includes several input fields: 'Line' (a dropdown), 'Number Called in on...' (containing '088211257790'), 'NAME' (a dropdown), 'COMPANY' (a text field), 'Office Number' (a text field), 'Mobile' (a text field), 'Spoke to:' (a dropdown), 'Caller Category' (a dropdown), 'Product Category' (a dropdown), and a large 'Notes' text area. At the bottom right, there are 'Cancel' and 'OK' buttons.

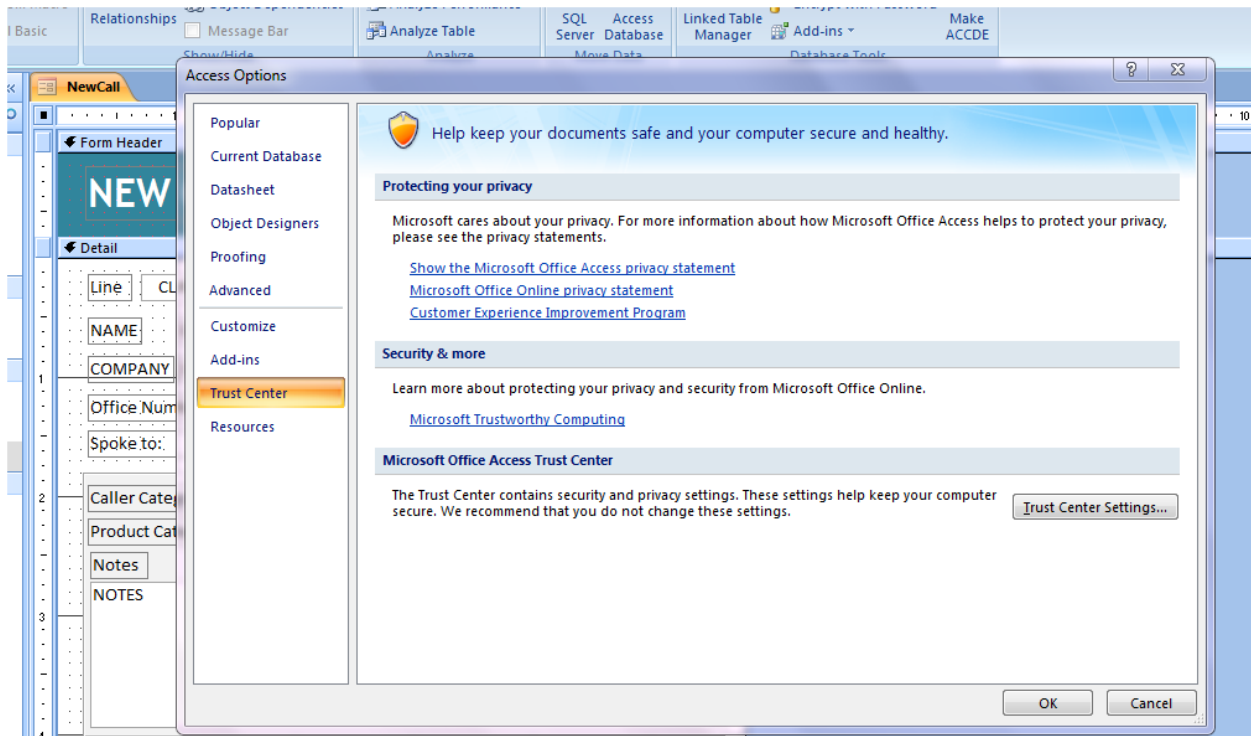
- I have prepared this database to reuse existing MS Access window if there's any. And if not, it will open a new one. If the database is currently opening NewCall form, it will first save it before reopening it. Of course, you must test it there to be sure

# PIMPHONY INTEGRATION UTILITY GUIDE

## Modifying FrontEnd Database

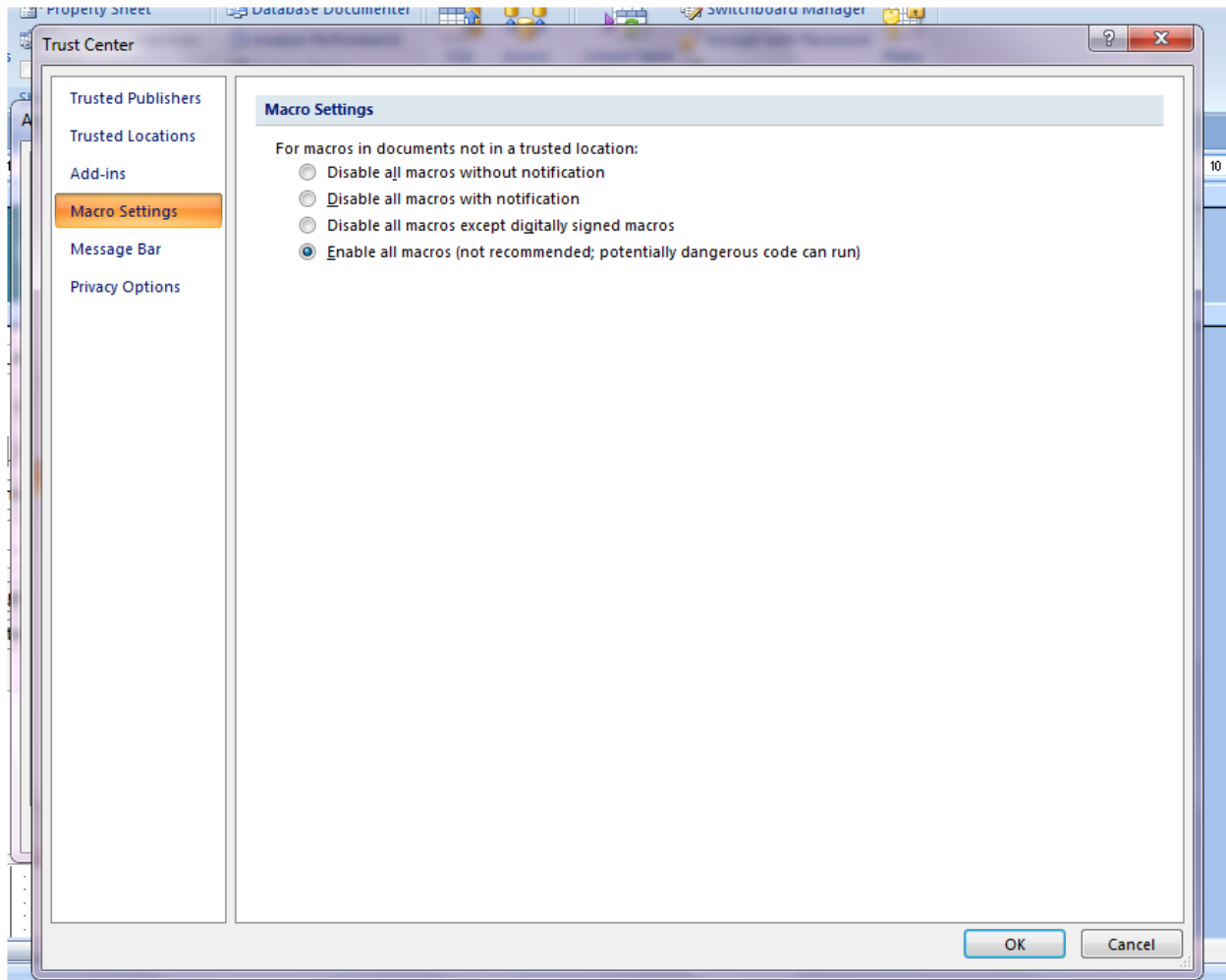
If I am not mistaken, the front end database is stored as the file PimphonyTest.accdb. You must first change the code that lies underneath Form NewCall to support receiving of caller number from Pimphony. To do this, first you must make sure that the Access Trust Center setting is modified as Follows :

Open Access Options, and click Trust Center:



Click Trust Center Settings, and choose Macro settings :

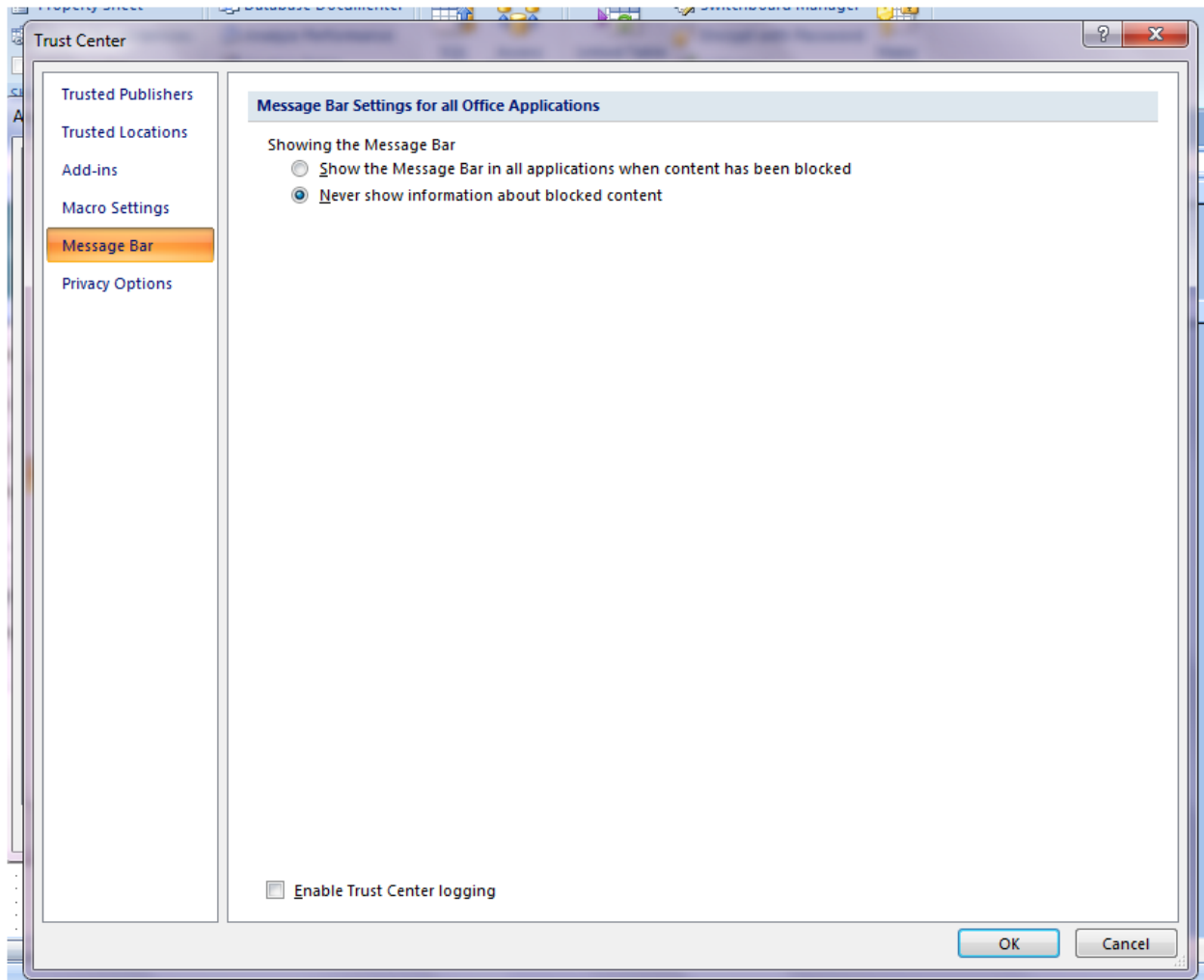
# PIMPHONY INTEGRATION UTILITY GUIDE



Choose : Enable all macros (not recommended; potentially dangerous code can run). Of course it's no problem. We did want our code to run, right? ;)

Also click Message Bar :

## PIMPHONY INTEGRATION UTILITY GUIDE

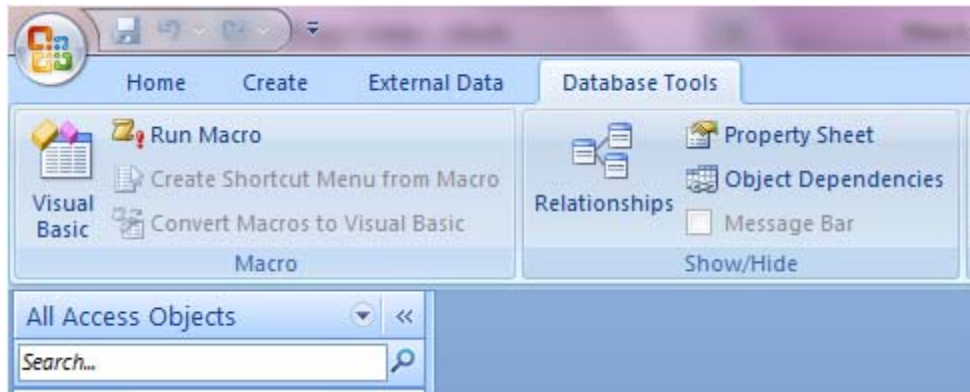


And choose Never show information about blocked content. This is for convenient only. As we already know we want our code to run, it's not informative to let Access warn us about it over and over again.

Now, that the Access environment already setup to allow us to run code and modify it (which, I think your environment already like this), we can proceed with the modification of the New Call form itself.

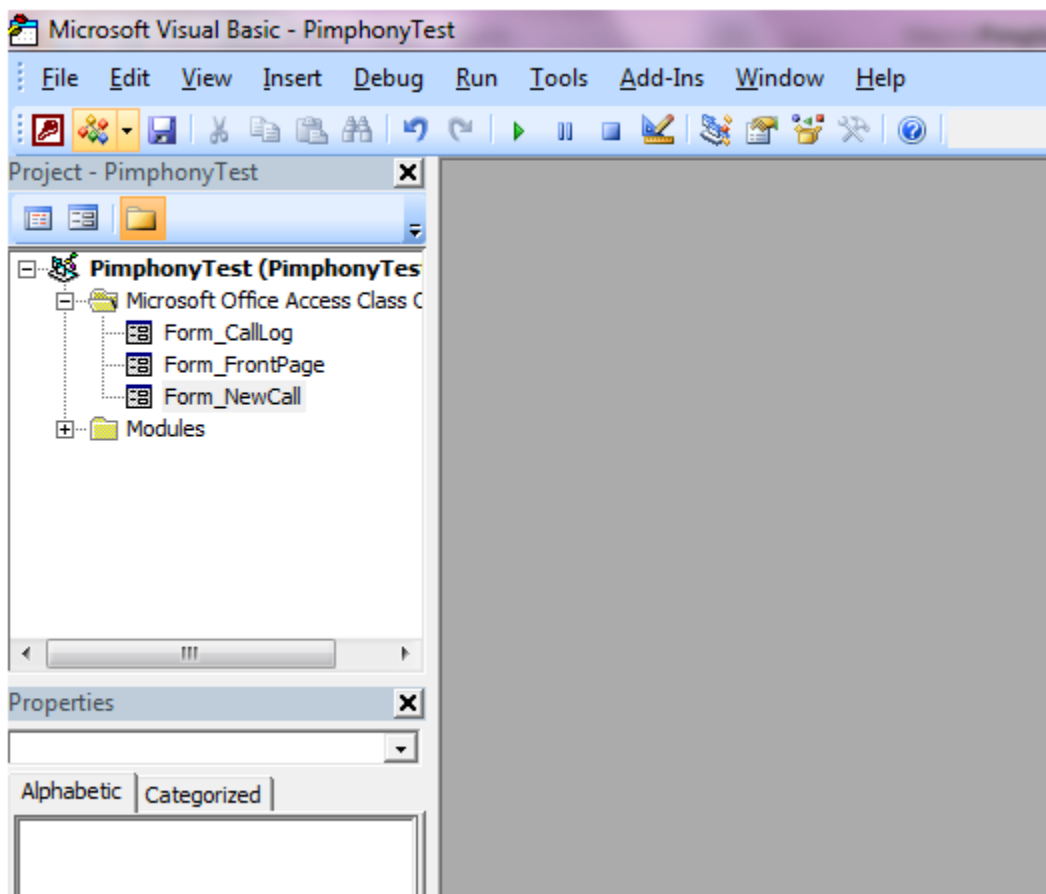
For this, Click Database Tools Ribbon bar :

## PIMPHONY INTEGRATION UTILITY GUIDE



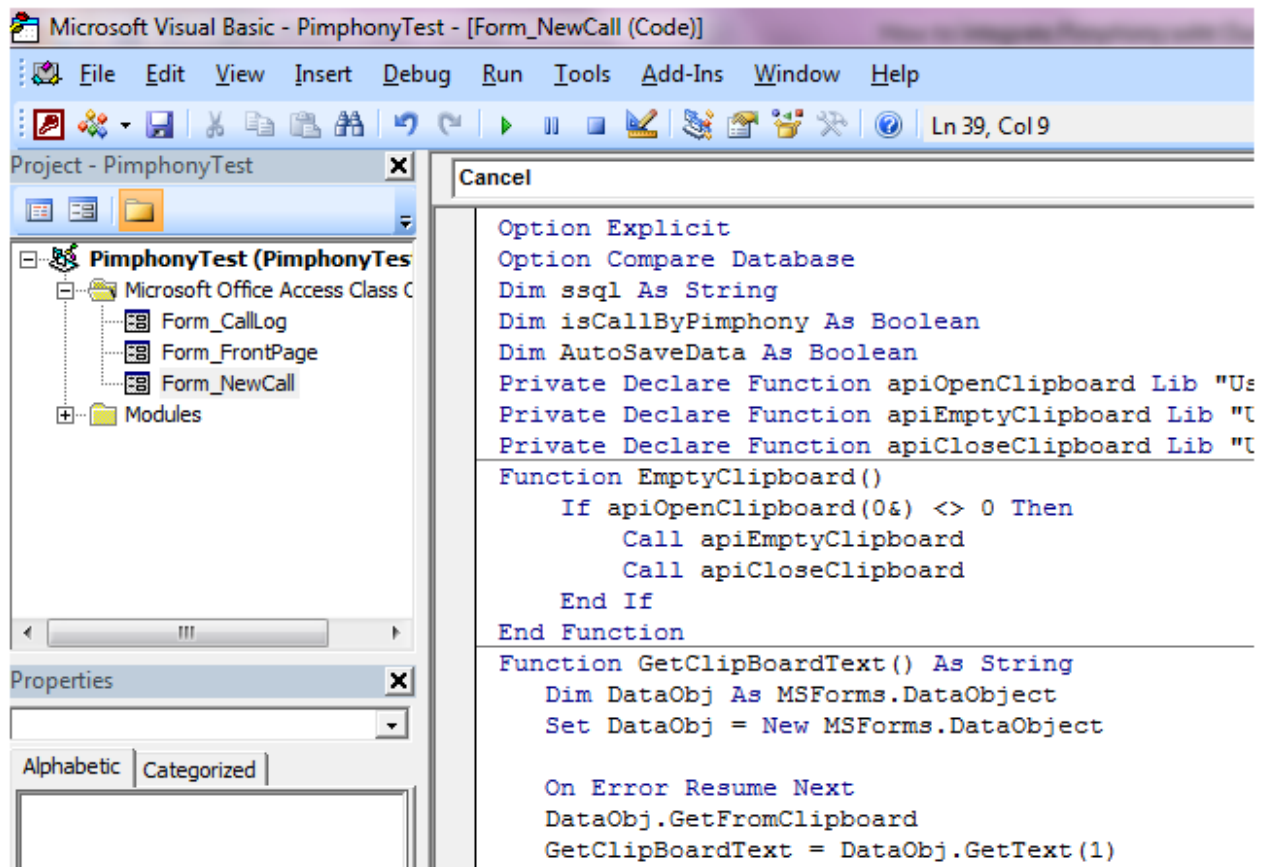
Klik that asthonorishingly large button, Visual Basic.

The Visual Basic Application code editor will open :



There you can see our form : Form\_NewCall. Double click on it, and you'll see the code editor as follows :

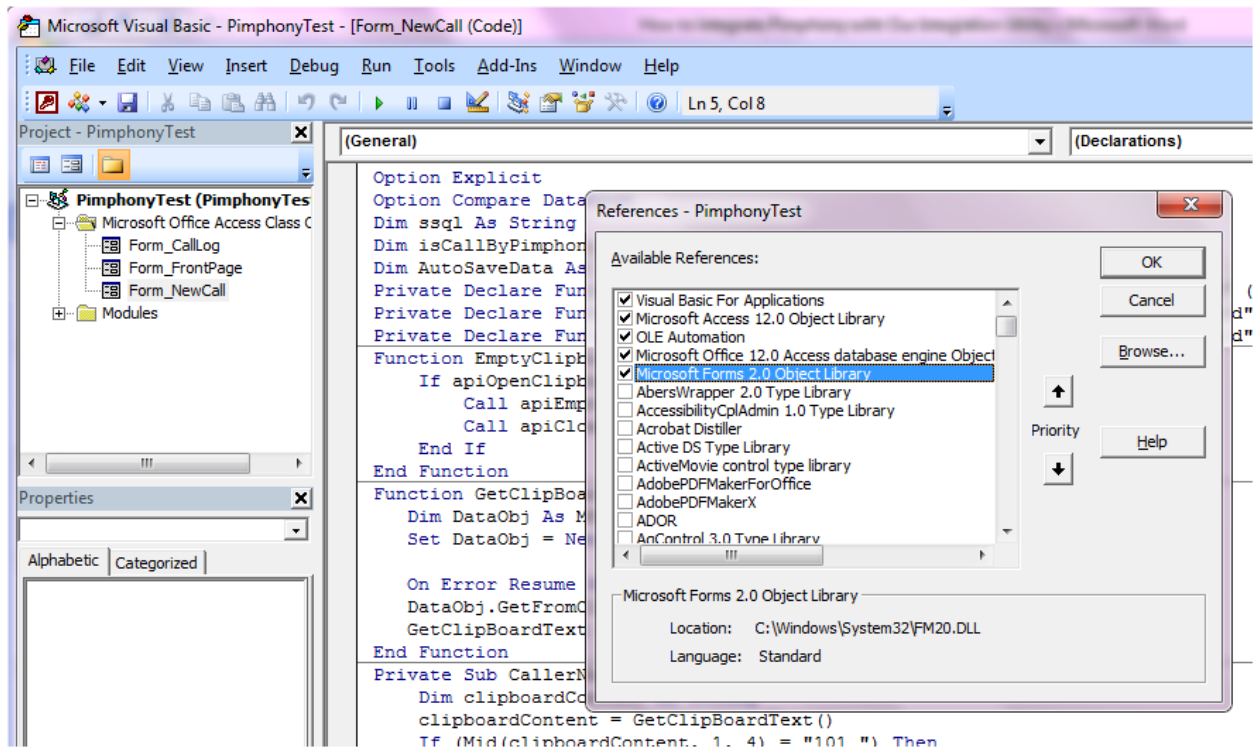
## PIMPHONY INTEGRATION UTILITY GUIDE



Of course this is my new code that support Pimphony integration. To replace this with the new one, select all code in this window (press CTRL+A), and DELL. Yep, it will erase all the old code. Now, paste to this editor, all the content from the file Form\_NewCall.txt that I upload in oDesk inbox message. You can just simply double click on the Form\_NewCall.txt, CTRL+A, Copy. And paste it here.

Before saving and closing the database, first, we must add reference to this database to MS Forms 2.0. I utilize windows Clipboard, to transfer phone number from Pimphony Integration Utility into MS Access Database. To do the reference, when you still in Code Editor, choose Menu Tools->References :

# PIMPHONY INTEGRATION UTILITY GUIDE



In this screenshot you can see Microsoft Forms 2.0 Object Library is selected. In the old database, there isn't any reference to this. To link the reference, klik Browse and got to folder C:\Windows\System32. Select the file FM20.DLL as shown below:

The screenshot shows the 'References - PimphonyTest' dialog box. The 'Add Reference' sub-dialog is open, displaying the 'System32' folder. The file 'FM20.DLL' is selected in the list. The 'File name' field contains 'FM20' and the 'Files of type' dropdown is set to 'Type Libraries (\*.olb;\*.tlb;\*.dll)'. The 'Open' button is highlighted.

Done.

Thanks,  
Eko



# PIMPHONY INTEGRATION UTILITY GUIDE