



MyPBX and Vtiger CRM Integration Configuration Guide

Version 1.0

Yeastar Information Technology Co. Ltd.

Overview

About MyPBX AMI

MyPBX provides an AMI (Asterisk Manager Interface) that allows a client program to connect to an Asterisk instance and issue commands or read events over TCP/IP stream. The default port for MyPBX AMI is TCP 5038.

About Vtiger CRM

The Vtiger CRM provides a module called **PBX Manager** Module that will allow users to use Vtiger to create calls and receive notifications on incoming calls using the Asterisk Call Manager API provided by Asterisk. This document describes configuration of Vtiger CRM to work with MyPBX.

Vtiger CRM download link:

https://wiki.vtiger.com/index.php/Installation_with_EXE

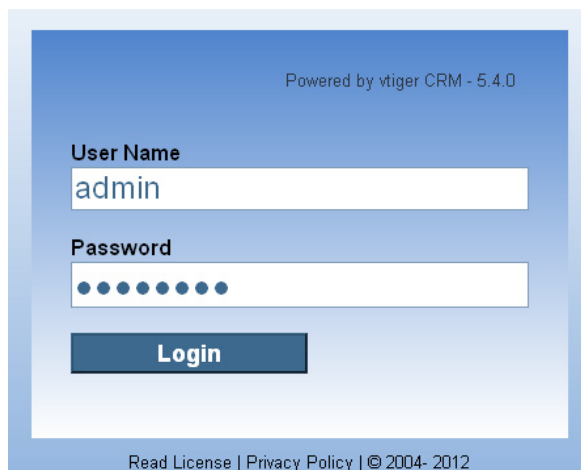
In this document, we will guide you to achieve the following features:

1. Click to call. Make calls using MyPBX extension by clicking on phone numbers (links) within Vtiger.
2. Incoming call popup. Provides notification within Vtiger giving details of the caller when a call comes to the user.
3. View all past calls under **PBX Manager** module.

Configuring Vtiger CRM

NOTE: in this guide, we shall install Vtiger CRM on a windows host. You can also install it on a Linux host. Once you are done with Vtiger CRM installation, you will have to connect with MyPBX.

1. Login Vtiger CRM with the username **admin** and the password you selected during the installation.



Powered by vtiger CRM - 5.4.0



User Name
admin

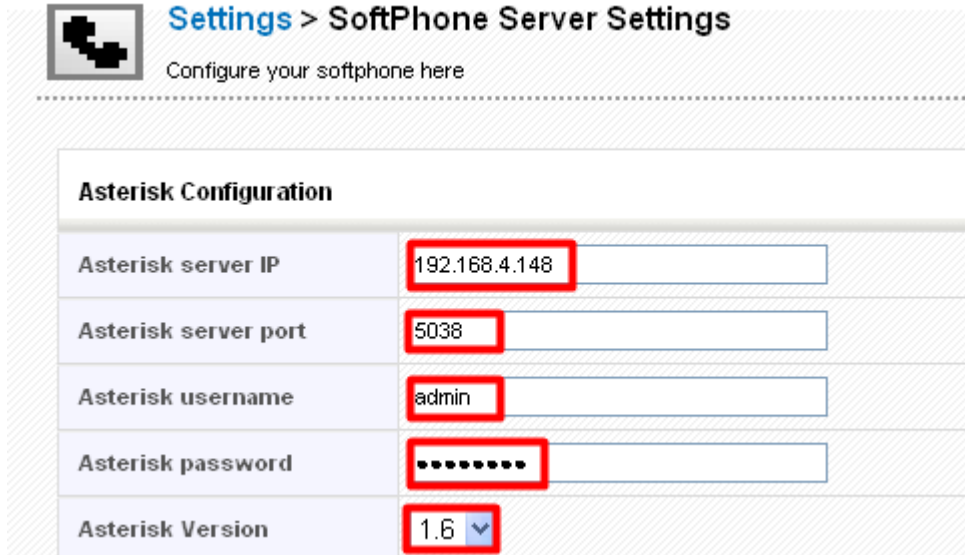
Password
●●●●●●●●

Login

Read License | Privacy Policy | © 2004- 2012


Figure 1

- Go to Settings Icon  > **CRM Settings** > **Module Manager**.
- Click on Settings Icon  next to **PBX Manager** module.
- You will be redirected to Settings page where you have to provide soft phone configuration details.




Asterisk Configuration	
Asterisk server IP	192.168.4.148
Asterisk server port	5038
Asterisk username	admin
Asterisk password
Asterisk Version	1.6

Figure 2

- Asterisk server IP : MyPBX IP, in this example: 192.168.4.148
 - Asterisk sever port: 5038
 - Asterisk username: MyPBX username
 - Asterisk password: MyPBX password
 - Astersik version: MyPBX uses Asterisk 1.6, here we choose 1.6
- Click on the button  to save the configuration.

Setting User Extensions

Vtiger CRM allows we to create MyPBX extensions for each user , which will let us call or receive call for that user. The following is steps to set User Extensions in Vtiger CRM.

- Login as any user and go to Settings Icon  , click on the **My Preferences** link on the top right of the page. On this page you will find a block labeled **Asterisk Configuration**.
- Asterisk Extension:** MyPBX extension, here take 705 for example.
- Receive Incoming Calls:** check the box, **yes** for enabling incoming calls.

7. Asterisk Configuration

Asterisk Extension	705	Receive Incoming Calls	<input checked="" type="checkbox"/>	Save or Cancel
--------------------	-----	------------------------	-------------------------------------	----------------

Figure 3

Click to Call

Till now, we have finished all the configurations on Vtiger CRM, but it is not over yet. There are steps we should do to make it work with MyPBX.

1. Edit the file AsteriskClass.php. Since we install Vtiger on Windows, so here we will find the file in the path:

```
\\vtigercrm-5.4.0\apache\htdocs\vtigerCRM\modules\PBXManager\utils\
AsteriskClass.php
```

The file on Linux will be found in:

```
www/html/Vtigercrm/modules/PBXManager/utils/AsteriskClass.php
```

In order to find the correct context, we will change the context according to the contexts in MyPBX. For example, context for extension 705 on MyPBX is [DLPN_DialPlan705]. Therefore, we will firstly get the extension number using the sentence “`$extensionnumber = split('V',$from);`” Then change the context with “`$context= "DLPN_DialPlan".$extensionnumber[1];`”

- Add the two sentences:


```
$extensionnumber = split('V',$from);
$context= "DLPN_DialPlan".$extensionnumber[1];
```
- Comment the section “switch” to disable it.
- Modified section of the file:


```
$extensionnumber = split('V',$from);
$context= "DLPN_DialPlan".$extensionnumber[1];
/*
switch($typeCalled){
case "SIP":
$context = " from-internal";
break;
case "PSTN":
$context = " ;//outbound-dialing";
break;
default:
$context = " ";
}*/
```
- Save the file.

2. Make an outgoing call to test.

- Choose one number link to click, once clicked, it will display a popup notifying the outgoing call being made.

First Name	Last Name	Title	Organization Name	Email	Office Phone
jerry	z		--		702
ada	chan		--		703

Figure 4



Figure 5

- The extension you saved in **User Preferences** will start getting a call. In this guide, 705 will ring.
- Pick up the call (705), a call will be made to the number clicked in Vtiger CRM (here is 702).
- Check the call log in PBX Manager.

Call From	Call To	Time Of Call	Status
<input type="checkbox"/> Administrator	jerry z...	2012-12-28 01:08:19	outgoing
<input type="checkbox"/> Administrator	jerry z...	2012-12-28 01:14:56	outgoing
<input type="checkbox"/> Administrator	jerry z...	2012-12-28 01:32:43	outgoing
<input type="checkbox"/> Administrator	ada chan...	2012-12-28 01:36:35	outgoing

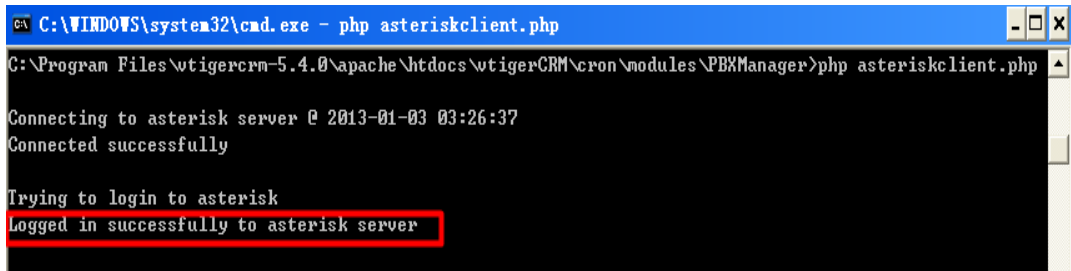
Figure 6

Incoming call popup

To receive incoming calls in Vtiger, we need to modify configuration file in Vtiger. The following is steps to achieve this feature pop-up call.

1. Modify file AsteriskClient.php
 - Find the configuration file AsteriskClient.php in the path:
 \Vtigercrm-5.4.0\apache\htdocs\VtigerCRM\cron\modules\PBXManager
 On Linux, the file will be found in:
 /var/www/html/Vtigercrm/cron/modules/PBXManager/AsteriskClient.php
 - Open the file and add a function in the file: **handleIncomingCalls**
 - Invoke the function **handleIncomingCalls** in the function **Asterisk_handleEvents**. Please refer to the **Appendix**.
 - In the function **asterisk_handleResponse4** , add “else if(\$mainresponse['Event'] == 'Dial' && \$mainresponse['SubEvent'] == 'Begin'){ }” behind
 “else if(\$mainresponse['Event'] == 'OriginateResponse'){ }”. For the detail, please refer to the **Appendix**.
 - Please refer to the modified section of AsteriskClient.php in the **Appendix**.

- Save the file.
- Enter the directory where the file AsteriskClient.php in with the CLI or terminal. Here we will use a Command prompt Window to enter the directory: \\vtigercrm-5.4.0\apache\htdocs\vtigerCRM\cron\modules\PBXManager
Input the command: **phpAsteriskClient.php** to execute the file. If it succeeds, you will see the picture as below:



```
C:\WINDOWS\system32\cmd.exe - php asteriskclient.php
C:\Program Files\vtigercrm-5.4.0\apache\htdocs\vtigerCRM\cron\modules\PBXManager>php asteriskclient.php

Connecting to asterisk server @ 2013-01-03 03:26:37
Connected successfully

Trying to login to asterisk
Logged in successfully to asterisk server
```

Figure 7

For Linux System, you need use terminal to do the operation.

2. Make a call to the user extension to test. Here we will call from 702 to 705. Once the call is set up, the users will get a pop-up notifying the caller name (if the number is present in Vtiger CRM) and the number.

Incoming Call

Caller Information

Number : 702
Name : 702

Information from Vtiger

jerry z [Contacts]

Figure 8

[Appendix]**Modified section of AsteriskClient.php**

```

function asterisk_handleEvents($asterisk, $adb, $version="1.4") {
    $fnEntryTime = time();
    //values of flag for asteriskincomingevents(-1 for stray calls, 0 for incoming calls,
    1 for outgoing call)
    Do
        $mainresponse = $asterisk->getAsteriskResponse();

        if(!empty($mainresponse)) {
            $state = ($version == "1.6")? "ChannelStateDesc" : "State";

            if(asterisk_handleResponse1($mainresponse, $state, $adb)) {
                if(asterisk_handleResponse2($mainresponse, $adb, $asterisk,
                $state)) {
                    if(handleIncomingCalls($asterisk, $adb, $version)) /* Invoke
                    the function handleIncomingCalls*/
                        //if(asterisk_handleResponse3($mainresponse, $adb,
                    $asterisk)){
                            // Proceed if previous event could not be handled.
                        }
                    }
                } else {
                    // No more response to consume
                    break;
                }
            } while(true);

            return false;
        }
}

function asterisk_handleResponse4($mainresponse, $adb, $asterisk, $state) {
    $appdata = $mainresponse['AppData'];

    $uniqueid = $channel = $callerType = $extension = null;
    $parseSuccess = false;

    if(
        $mainresponse['Event'] == 'Newexten' && (strstr($appdata,
        "__DIALED_NUMBER") || strstr($appdata, "EXTTOCALL"))
    ){

```

```

$uniqueid = $mainresponse['Uniqueid'];

$channel = $mainresponse['Channel'];
$splits = explode('/', $channel);
$callerType = $splits[0];

$splits = explode('=', $appdata);
$extension = $splits[1];

$parseSuccess = true;
} else if($mainresponse['Event'] == 'OriginateResponse'){
    //if the event is OriginateResponse then its an outgoing call and set the flag to
    1, so that AsteriskClient does not pick up as incoming call
    $uniqueid = $mainresponse['Uniqueid'];
    $adb->pquery("UPDATE vtiger_asteriskincomingevents set flag = 1 WHERE
    uid = ?", array($uniqueid));
    }else if($mainresponse['Event'] == 'Dial'&& $mainresponse['SubEvent'] ==
    'Begin'){
        $uniqueid = $mainresponse['UniqueID'];

        $channel = $mainresponse['Channel'];
        $splits = explode('/', $channel);
        $callerType = $splits[0];
        $extension = $mainresponse['Dialstring'];
        $parseSuccess = true;
    }
    /* Add else if($mainresponse['Event'] == 'Dial'&& $mainresponse['SubEvent'] ==
    'Begin'){ } */

    /*else if($mainresponse['Event'] == 'Hangup'){
    $uniqueid = $mainresponse['Uniqueid'];

    $channel = $mainresponse['Channel'];
    $splits = explode('/', $channel);
    $callerType = $splits[0];

    $extension = $mainresponse['CallerIDNum'];

    $parseSuccess = true;
    }
    */

    if($parseSuccess) {

```



```

if(checkExtension($extension, $adb)) {

    $sql = "UPDATE vtiger_asteriskincomingevents SET to_number=?,
callertype=?, timer=?, flag=? WHERE uid=?";
    $adb->pquery($sql, array($extension, $callerType, time(), 0, $uniqueid));
    echo "UPDATE vtiger_asteriskincomingevents SET
to_number=$extension, callertype=$callerType, timer=".time().", flag=0 WHERE
uid=$uniqueid\n";
    //echo "xxxxxxxxxxxxxxxxx          ".$adb->num_rows($result)."\n";
    $callerinfo = $adb->pquery("SELECT from_number,from_name FROM
vtiger_asteriskincomingevents WHERE uid = ?",array($uniqueid));
    echo $uniqueid."          ffffffffffffffffffff
".$adb->num_rows($callerinfo)."\n";
    if($adb->num_rows($callerinfo) > 0){
        $callerNumber = $adb->query_result($callerinfo, 0, "from_number");
        $callerName = $adb->query_result($callerinfo, 0, "from_name");

        if(empty($callerNumber) || $callerNumber == '0') {
            // We don't have the information who is calling, could happen in
Asterisk 1.4 (when call is made to Queue)
            // Let us defer the popup show for next Event: Link
            $sql = "UPDATE vtiger_asteriskincomingevents SET flag=?
WHERE uid=?";
            $adb->pquery($sql, array(-1, $uniqueid));
        } else {
            $query = "INSERT INTO vtiger_asteriskincomingcalls (refuid,
from_number, from_name, to_number, callertype, flag, timer) VALUES(?,?,?,?,?,?)";
            $adb->pquery($query,array($uniqueid, $callerNumber,
$callerName, $extension, $callerType, 0, time()));
        }
    }
}

return false;
}

return true;
}

```

Add function handleIncoming Calls:

```
function handleIncomingCalls($asterisk, $adb, $version="1.4"){
```

```

$response = $asterisk->getAsteriskResponse();
if(empty($response)){
    return false;
}

$callerNumber = "Unknown";
$callerName = "Unknown";
if($version == "1.6"){
    $state = "ChannelStateDesc";
}else{
    $state = "State";
}

if(($response['Event'] == 'Newstate' || $response['Event'] == 'Newchannel') &&
($response[$state] == 'Ring' || $response[$state] == 'Ringing')){

    if(!empty($response['CallerID'])){
        $callerNumber = $response['CallerID'];
    }elseif(!empty($response['CallerIDNum'])){
        $callerNumber = $response['CallerIDNum'];
    }
    if(!empty($response['CallerIDName'])){
        $callerName = $response['CallerIDName'];
    }
    $unique_id = $response['Uniqueid'];
    while(true){
        $response = $asterisk->getAsteriskResponse();
        if(($response['Event'] == 'Dial') && ($response['CallerIDNum'] ==
$callerNumber) && ($response['UniqueID'] == $unique_id)){

            $dest_unique_id = $response['DestUniqueID'];
            while (true) {
                $response = $asterisk->getAsteriskResponse();
                if (($response['Event'] == 'Newstate') &&
($response['ChannelStateDesc'] == 'Ringing') && ($response['Uniqueid'] ==
$dest_unique_id)) {
                    if(strpos($response['Channel'], $callerNumber)){
                        $temp = explode("/", $response['Channel']);
                        $callerType = $temp[0];
                    }
                    break;
                }
            }
        }
        break;
    }
}

```

```
    }  
  }  
}else{  
  return false;  
}  
}
```

<The end>