vtlib – vtiger development library

version 1.1

Table of Contents

About

vtlib aims at providing easier development environment for vtiger. In this version API's are provided which lets you create module easily for vtiger.

Requirements

vtiger 5.0.4 should be installed.

NOTE: Source directory in this document refers, vtiger CRM source installation.
If you have used vtiger CRM bundled installation like, .exe or .bin, it will be located in
<vtigercrm_install_directory>\apache\htdocs\vtigerCRM

Applying Customization

1. Take backup of your vtiger CRM Source directory.
2. Unpack the vtlib-x.y.zip into your vtiger CRM source directory.

Files and Folder:

Existing file changes		
include	export.php	Hooks added to support record export for modules.
modules/Import	ImportStep1.php	Hooks added to support record import into modules.
	ImportStep2.php	
	ImportStep3.php	
	ImportSteplast.php	
	ImportStepundo.php	
	UsersLastImport.php	
Smarty/templates	Buttons_List.tpl	Hook provided to enable import and export buttons for custom modules.
	Buttons_List1.tpl	
New file additions		
modules/Payslip		Example modules for testing vtlib scripts.
modules/EmailScanner		

Test Scripts

vtlib.Test.html	Provides link to execute the following scripts
vtlib.Test.Create.Menu.Item.php	Example to create menu item
vtlib.Test.Create.Module.Blocks.php	Example to show module creation and block for it.
vtlib.Test.Create.Module.Fields.php	Example to show module creation and fields for it.
vtlib.Test.Create.Module1.php	Payslip Module Creation (modules/Payslip dir is already packaged) Make the following entry in include/language/en_us.lang.php after script execution $app_strings = Array('Payslip' => 'PaySlip', ...

vtlib.Test.Create.Module2.php	EmailScanner Module Creation (modules/EmailScanner dir is already packaged) Make the following entry in include/language/en_us.lang.php after script execution $app_strings = Array('EmailScanner => 'Email Scanner', ...

Testing Module Creation

1. After unpacking the vtlib zip file, open http://localhost/vtigercrm/vtlib.Test.html
2. Click on Create Payslip Module to test creation of Payslip Module
3. Click on Create EmailScanner Module to test creation of EmailScanner Module

NOTE: When you create module as said above, you will see blank links added under Tools menu. Kindly edit include/language/en_us.lang.php and update $app_strings as shown below:

```
$app_strings = Array (
'Payslip' => 'PaySlip',
'EmailScanner' => 'EMAIL SCANNER',
...
```

After this you will be able to see Tools » Payslip and Tools » EMAIL SCANNER

API Usage

Given below are examples on how to use the API's provided by vtlib.

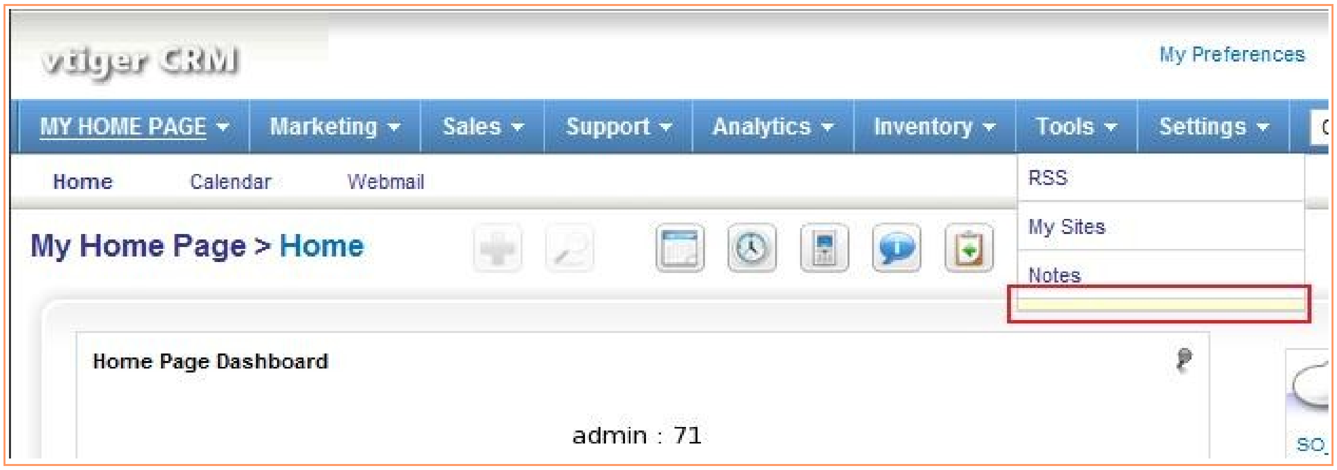
Creating Tab (Menu Item)

```
include_once('vtlib/Vtiger/Tab.php');
Vtiger_Tab::create('<NEWMODULE>', '<NEWMODULE_LABEL>', '<MENUNAME>');
```

This API will create new menu item which serves as entry point for the module.

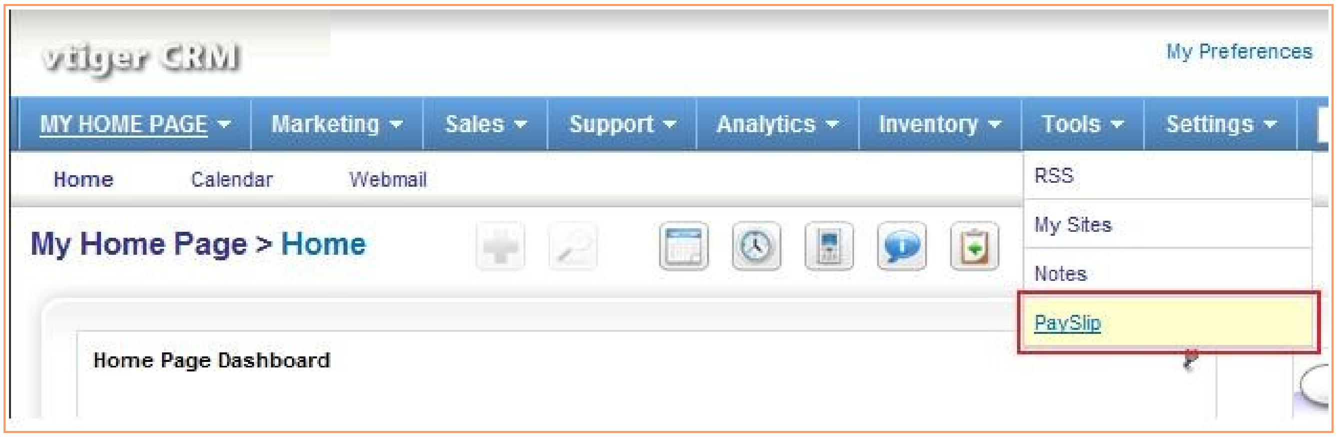
Example:

```
Vtiger_Tab::create('Payslip', 'Payslip', 'Tools');
```



NOTE: You will need to put the <NEWMODULE_LABEL> mapping in the language file (include/language/en_us.lang.php) to see the affect.

```
$app_strings = Array (
'Payslip' => 'PaySlip',
...
```



Creating Block

```
include_once('vtlib/Vtiger/Block.php');
Vtiger_Block::create('<MODULE>', '<BLOCK_LABEL>');
```

This API will create block for the specified <MODULE>.

Example:

```
Vtiger_Block::create('Leads', 'LBL_VTLIB_INFORMATION');
```

NOTE: You will need to put the <BLOCK_LABEL> mapping in the module language file (modules/<MODULE>/language/en_us.lang.php) to see the affect.

For the above example, add the mapping in modules/Leads/language/en_us.lang.php

```
$mod_strings = Array (
'LBL_VTLIB_INFORMATION' => 'vtlib Information',
...
```

NOTE2: The block will not be displayed until we associate one field to it.

Creating Field

```
include_once('vtlib/Vtiger/Field.php');
$field1 = new Vtiger_Field();

// Populate Field Information

$field1->set('module',       '<MODULE>')
      -> set('columnname',    '<TABLE COLUMN NAME>')
      -> set('tablename',     '<TABLE NAME>')
      -> set('columntype',    '<TABLE COLUMN TYPE>')
      -> set('generatedtype', '<STANDARD (=1) OR CUSTOM (=2) TYPE>')
      -> set('uitype',        <UI TYPE>)
      -> set('fieldname',     '<FIELD NAME>')
      -> set('fieldlabel',    '<FIELD LABEL ON UI>') // Need language file entry
      -> set('readonly',       '<READONLY (=0) orelse (1)>')
      -> set('presence',       '0')
      -> set('selected',       '0')
      -> set('maximumlength',  '<MAXIMUM DATA LENGTH (=100)>')
      -> set('sequence',       null) // Will be set dynamically.
      -> set('typeofdata',     '<TYPE OF DATA>')
      -> set('quickcreate',    '<AVAILABLE IN QUICK CREATE FORM(=0) orelse(=1)>')
      -> set('block',          null) // Set blockid and make blocklabel null
      -> set('blocklabel',     '<BLOCK LABEL to which field needs to attached>')
      -> set('displaytype',    '<DISPLAY TYPE>')
      -> set('quickcreatesequence', null) // Will be set dynamically
      -> set('info_type',      '<TYPE OF INFORMATION (BAS=Basic,ADV=Advanced)>');

// Create Field

$field1->create();
```

The above API will let you create vtiger field and assoicate it to the module block.
You will have to provide all the information described above.

NOTE: During vtiger field creation, the columnname existence is checked in the tablename specified. If it does not exists then tablename is alter and columnname (columntype) is created.

You can create table before creating field & provide the tablename, columnname to use this table.

```
include_once('vtlib/Vtiger/Utils.php');
Vtiger_Utils::CreateTable('<TABLENAME>', '(<COLUMNNAME COLUMNTYPE>)');
```

This API will let us create table with tablename and given column conditions.

Example: To create a new field and associate to custom block in Leads module.

```
include_once('vtlib/Vtiger/Field.php');

$field1 = new Vtiger_Field();
$field1-> set('module',     'Leads')
      -> set('columnname',    'vtlib_leads')        // New column
      -> set('tablename',     'vtiger_leaddetails') // Existing table
      -> set('columntype',    'varchar(255)')
      -> set('generatedtype', '1')
      -> set('uitype',        2)
      -> set('fieldname',     'vtlibldname')
      -> set('fieldlabel',     'vtlibLeadName')
                        // Entry needed in modules/Leads/language/en_us.lang.php
      -> set('readonly',       '1')
      -> set('presence',       '0')
      -> set('selected',       '0')
      -> set('maximumlength',  '100')
      -> set('sequence',       null)
      -> set('typeofdata',     'V~M')
      -> set('quickcreate',    '1')
      -> set('block',          null)
      -> set('blocklabel',     'LBL_VTLIB_INFORMATION')
      -> set('displaytype',    '1')
      -> set('quickcreatesequence', null)
      -> set('info_type',      'BAS');

$field1->create();
```

After executing this script, make sure to check the mapping in modules/Leads/language/en_us.lang.php for fieldlabel (vtlibLeadName) used above.

If you have associated the field with a new table then you will need to update: *modules/Leads/Leads.php* $tab_name and $tab_name_index Arrays

```
   var $tab_name = Array( ..., '<NEW_TABLENAME>');
   var $tab_name_index = Array( ..., '<NEW_TABLENAME>' => '<ID_COLUMN>');
```

Creating Custom View

```
include_once('vtlib/Vtiger/CustomView.php');
Vtiger_CustomView::create('<MODULE>', '<FILTER NAME>');
```

This API will let you create custom view (filter) for the required module. Once the filter is created you need to associate module fields to it. The module field instance need to be populated as explained in previous section.

```
$cv = new Vtiger_CustomView('<MODULE>', '<FILTER NAME>');
$cv->addColumn($Vtiger_Field_Instance);
```

Example:
Create custom view for Leads module and associating the field created in last section.

```
include_once('vtlib/Vtiger/CustomView.php');
include_once('vtlib/Vtiger/Field.php');

Vtiger_CustomView::create('Leads', 'vtlibFilter');

$cv = new Vtiger_CustomView('Leads', 'vtlibFilter');
$field1-> set('module',         'Leads')
      -> set('columnname',    'vtlib_leads')
      -> set('tablename',     'vtiger_leaddetails')
      -> set('fieldname',     'vtlibldname')
      -> set('fieldlabel',    'vtlibLeadName')
      -> set('typeofdata',     'V~M');

$cv->addColumn($field1);
```

Enable/Disable Action

Import, Export and Merge are treated as Actions for a module. To enable or disable them, you can use the API given below.

```
include_once('vtlib/Vtiger/Tab.php');
Vtiger_Tab::disableAction('<MODULE>','<Action>');
Vtiger_Tab::enableAction('<MODULE>','<Action>');
```

Example:
To Disable Import and Enable Export for Leads module.

```
Vtiger_Tab::disableAction('Leads','Import');
Vtiger_Tab::enableAction('Leads', 'Export');
```

NOTE: This will enable or disable the action (tool) for all the existing profiles.

Creating New Module

This can be broken down into two steps:

1. Database setup
2. Module directory setup

Database setup

Using the above API's explained above you will be able to create new modules for vtiger. The following are important steps that should be followed to get basic working module.

a) Create tab (entry point for module on UI)
b) Create required module table, module group table.
c) Create blocks for the module.
d) Create fields and associate it to blocks.
Set atleast one of the field as entity identifer.
e) Create custom view with required fields (make sure to create filter name All which is treated as special default filter)

NOTE: Please look at vtlib.Test.Create.Module1.php file to understand the steps explained above.

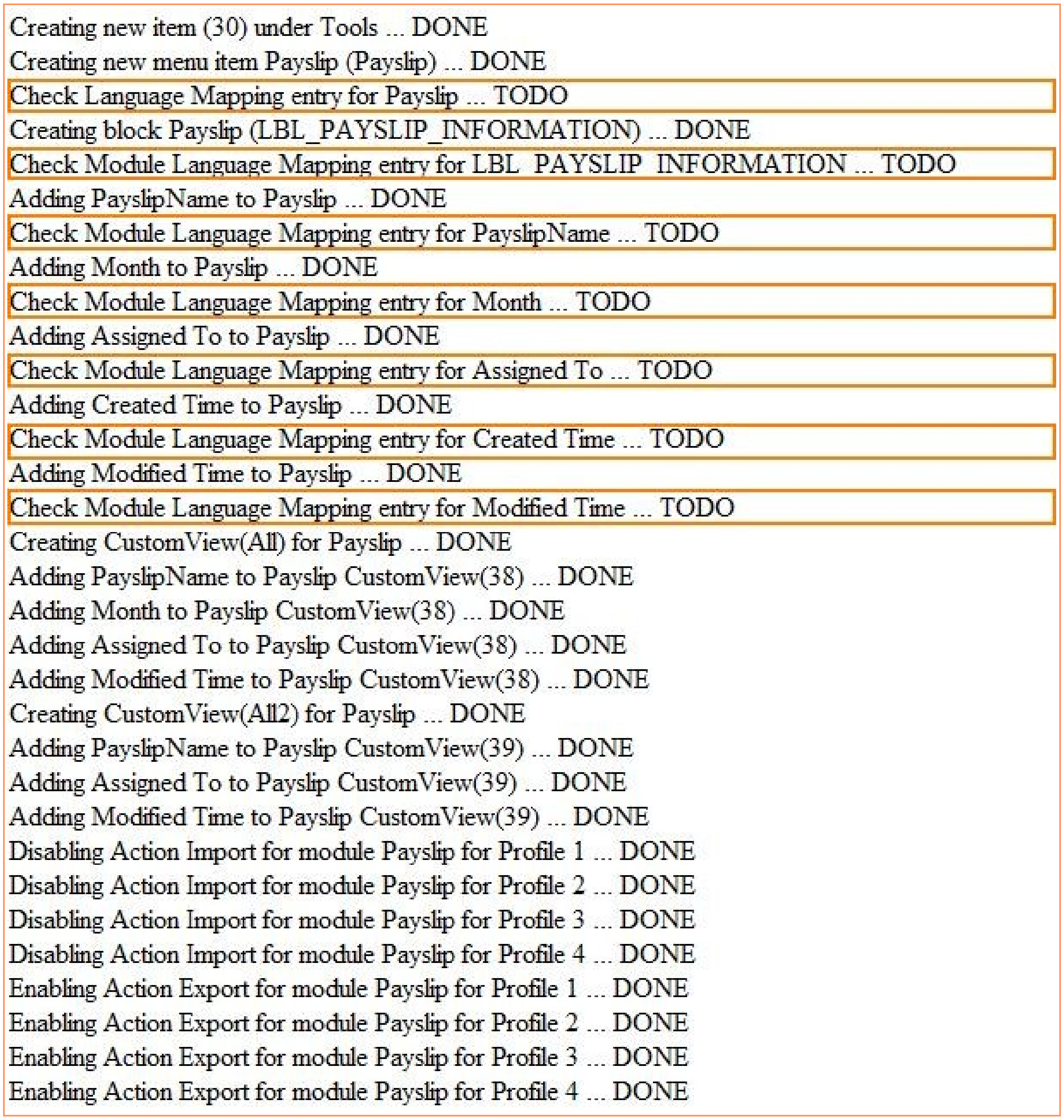
Module directory setup

vtlib directory contains skeleton ModuleDir which can be used as a template for new module that is created. It contains source files that needs to be changed accordingly.

1. Rename ModuleDir to <NewModuleName>
2. Rename ModuleDir/ModuleFile.php to <NewModuleName>.php
3. Rename ModuleDir/ModuleFileAjax.php to <NewModuleName>Ajax.php
4. Edit <NewModuleName>.php
   a) Rename Class ModuleClass to <NewModuleName>
   b) Update $table_name and $table_index (Module table name and table index column)
   c) Update $groupTable
   d) Update $tab_name, $tab_name_index
   e) Update $list_fields, $list_fields_name, $sortby_fields
   f) Update $detailview_links
   g) Update $default_order_by, $default_sort_order
   h) Update $required_fields
   i) Rename function ModuleClass to function <NewModuleName> [This is the Constructor Class]

Creating Payslip Module

When Create Payslip Module link is clicked from http://localhost/vtigercrm/vtlib.Test.html you will see the following:



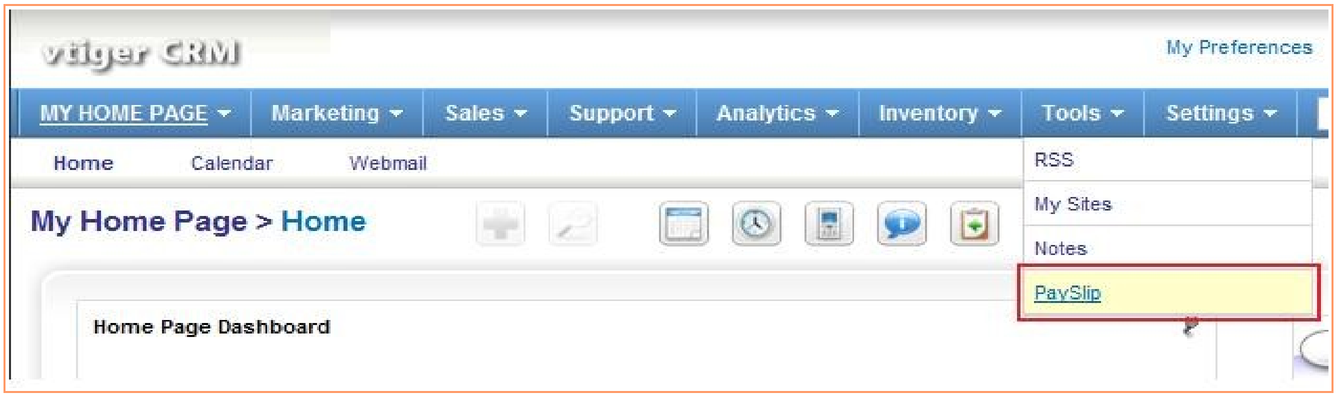
```
Creating new item (30) under Tools ... DONE
Creating new menu item Payslip (Payslip) ... DONE
Check Language Mapping entry for Payslip ... TODO
Creating block Payslip (LBL_PAYSLIP_INFORMATION) ... DONE
Check Module Language Mapping entry for LBL_PAYSLIP_INFORMATION ... TODO
Adding PayslipName to Payslip ... DONE
Check Module Language Mapping entry for PayslipName ... TODO
Adding Month to Payslip ... DONE
Check Module Language Mapping entry for Month ... TODO
Adding Assigned To to Payslip ... DONE
Check Module Language Mapping entry for Assigned To ... TODO
Adding Created Time to Payslip ... DONE
Check Module Language Mapping entry for Created Time ... TODO
Adding Modified Time to Payslip ... DONE
Check Module Language Mapping entry for Modified Time ... TODO
Creating CustomView(All) for Payslip ... DONE
Adding PayslipName to Payslip CustomView(38) ... DONE
Adding Month to Payslip CustomView(38) ... DONE
Adding Assigned To to Payslip CustomView(38) ... DONE
Adding Modified Time to Payslip CustomView(38) ... DONE
Creating CustomView(All2) for Payslip ... DONE
Adding PayslipName to Payslip CustomView(39) ... DONE
Adding Assigned To to Payslip CustomView(39) ... DONE
Adding Modified Time to Payslip CustomView(39) ... DONE
Disabling Action Import for module Payslip for Profile 1 ... DONE
Disabling Action Import for module Payslip for Profile 2 ... DONE
Disabling Action Import for module Payslip for Profile 3 ... DONE
Disabling Action Import for module Payslip for Profile 4 ... DONE
Enabling Action Export for module Payslip for Profile 1 ... DONE
Enabling Action Export for module Payslip for Profile 2 ... DONE
Enabling Action Export for module Payslip for Profile 3 ... DONE
Enabling Action Export for module Payslip for Profile 4 ... DONE
```

NOTE: Kindly edit include/language/en_us.lang.php and update $app_strings as shown below:

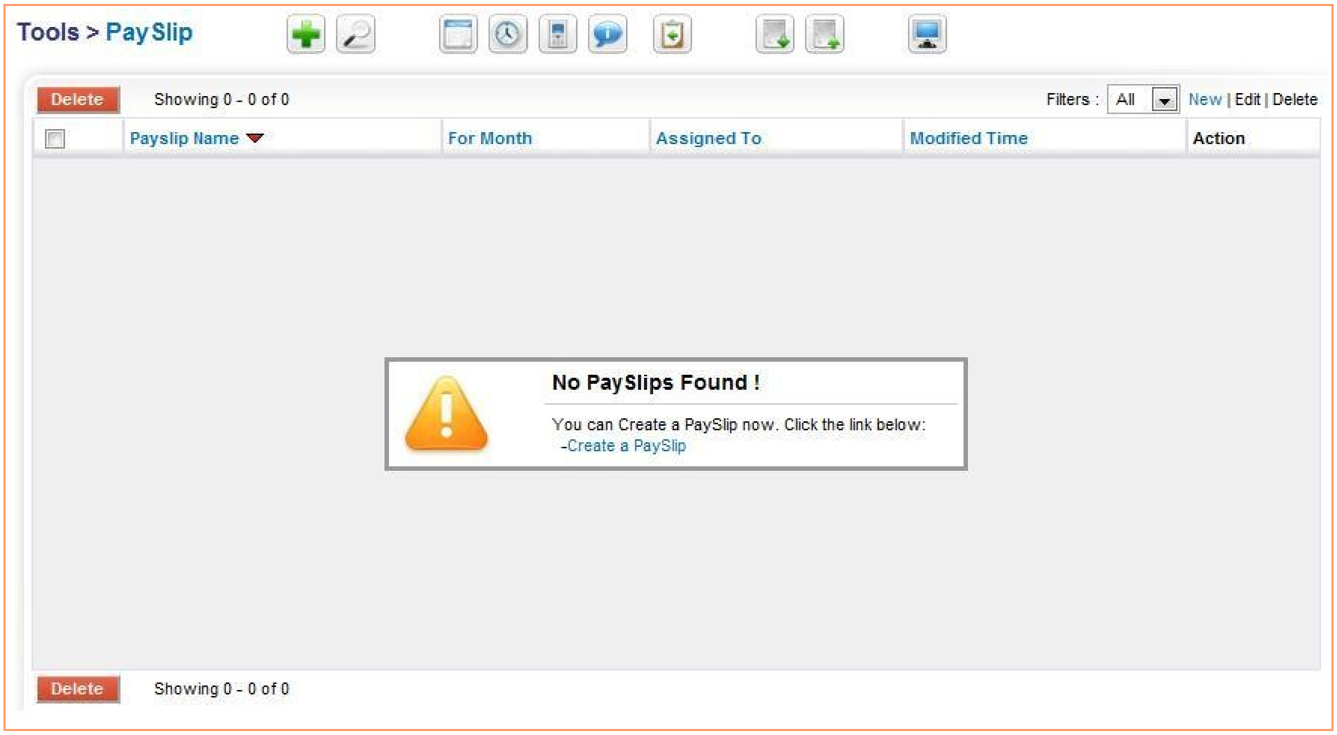
```
$app_strings = Array (
'Payslip' => 'PaySlip'
...
```

Module Entry Point

The module link appears under Tools Menu.



Default List view



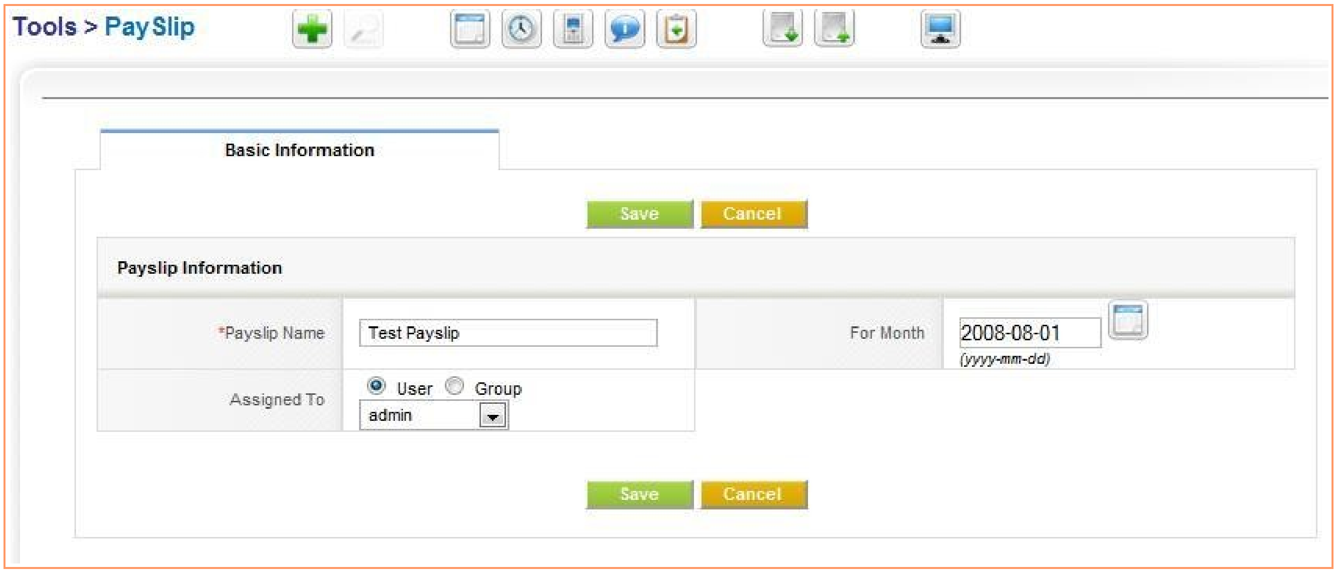
Showing 0 - 0 of 0

No PaySlips Found !

You can Create a PaySlip now. Click the link below:
-Create a PaySlip

Showing 0 - 0 of 0

Create view



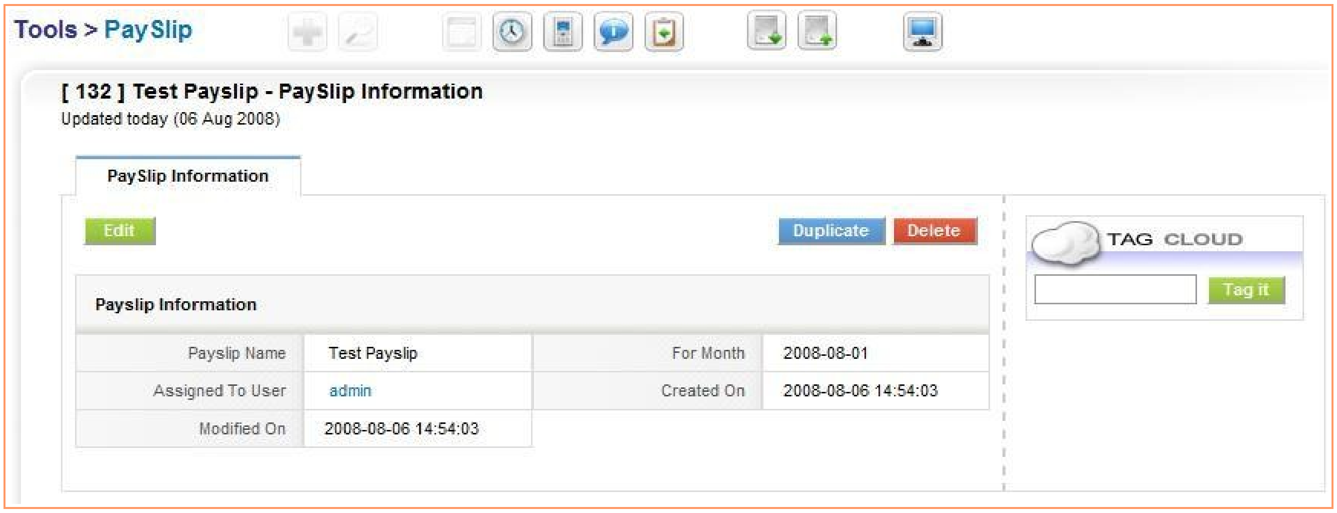
Tools > PaySlip

Basic Information

Payslip Information

*Payslip Name	Test Payslip	For Month	2008-08-01 (yyyy-mm-dd)
Assigned To	● User ○ Group admin		

Detail view



Tools > PaySlip

[132] Test Payslip - PaySlip Information

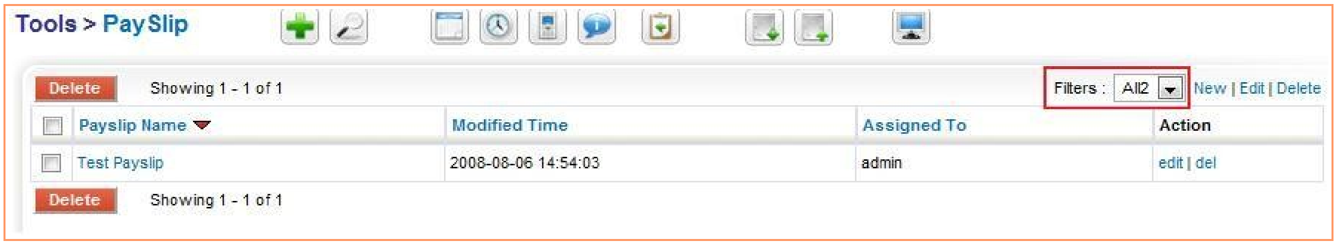
Updated today (06 Aug 2008)

PaySlip Information

Payslip Information

Payslip Name	Test Payslip	For Month	2008-08-01
Assigned To User	admin	Created On	2008-08-06 14:54:03
Modified On	2008-08-06 14:54:03		

List view



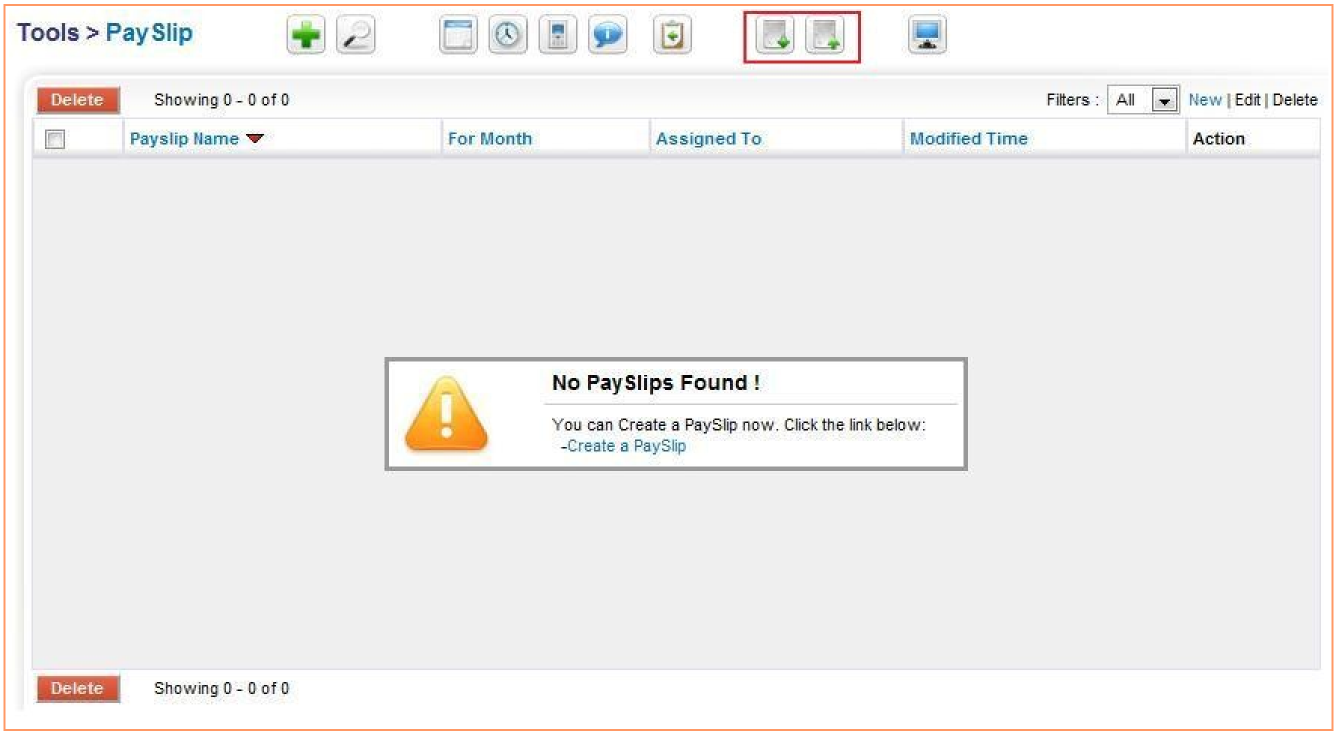
Tools > PaySlip

Showing 1 - 1 of 1

□	Payslip Name	Modified Time	Assigned To	Action
□	Test Payslip	2008-08-06 14:54:03	admin	edit | del

Showing 1 - 1 of 1

Export – Import Tool



No PaySlips Found !

You can Create a PaySlip now. Click the link below:

-Create a PaySlip