



# **BÁO CÁO**

## **HƯỚNG DẪN PHÁT TRIỂN MODULE**

### **TRÊN RVX**

Hà Nội, 08/2012

## A. Phần mở đầu

Trong thời gian vừa qua, dưới sự hướng dẫn tận tình của các anh, các chị trong công ty ViamiSoftware, chúng tôi, những sinh viên đang thực tập tại công ty đã được tiếp xúc và làm việc với phần mềm quản trị doanh nghiệp RVX Manager của công ty. Và sau thời gian học hỏi và tích lũy kinh nghiệm đầy quý giá đó, chúng tôi đã hoàn thành bản báo cáo “Hướng dẫn phát triển Module trên RVX Manager” dựa trên những kiến đã có. Bản báo cáo là tập hợp những kiến thức cơ bản nhất để có thể phát triển Module trên RVX Manager.

Qua đây chúng tôi xin cảm ơn Lãnh đạo, các anh, các chị trong công ty đã tạo điều kiện thuận lợi để chúng tôi có điều kiện hoàn thành Báo cáo này một cách hoàn thiện nhất.

### **Nhóm sinh viên thực hiện**

Nguyễn Khánh Hà  
Nguyễn Tiến Dũng  
Vũ Văn Thành Luân

## Mục lục

<b>A. Phần mở đầu .....</b>	<b>1</b>
<b>B. Giới thiệu tác giả .....</b>	<b>4</b>
<b>C. Mô tả khái quát về RVX.....</b>	<b>5</b>
<b>I. Giới thiệu chung.....</b>	<b>5</b>
<b>II. Lịch sử hình thành và phát triển .....</b>	<b>5</b>
<b>III. Công nghệ sử dụng.....</b>	<b>6</b>
1. PHP.....	6
2. Ext Js.....	6
3. Mysql .....	7
<b>IV. Khái quát các tính năng và độ ưu việt.....</b>	<b>8</b>
<b>V. Hướng dẫn cài đặt.....</b>	<b>9</b>
1. Cài đặt trên Windows .....	9
2. Cài đặt trên Linux .....	9
3. Sử dụng.....	10
<b>D. Tổng quan về module.....</b>	<b>11</b>
<b>I. Tổng quan .....</b>	<b>11</b>
1. Module .....	11
2. Model .....	11
3. Các file khác .....	12
<b>II. Cấu trúc các thành phần.....</b>	<b>12</b>
1. Module .....	12
2. Model .....	19
<b>E. Cách tạo 1 module đơn giản.....</b>	<b>29</b>
<b>I. Tạo và định nghĩa module mới .....</b>	<b>29</b>
<b>II. Tạo và định nghĩa model mới.....</b>	<b>30</b>
<b>III. Tạo report trên form.....</b>	<b>33</b>
<b>F. Tạo Model Cha, Con .....</b>	<b>36</b>
<b>I. Liên kết khóa ngoại.....</b>	<b>36</b>

II.	Tạo form Cha, Con.....	37
G.	Tạo module nâng cao .....	39
I.	File Controller .....	39
1.	Lớp Controller .....	39
2.	Lớp Model .....	41
3.	Lớp List .....	43
4.	Lớp Form .....	44
II.	Lọc List.....	44
III.	Lọc Combobox.....	47
IV.	Tạo report thống kê.....	48
V.	Tạo dashboard .....	50
H.	Tạo module kế thừa module đã có .....	51
I.	Tạo action.....	52
I.	Action trên Form.....	52
II.	Action trên List.....	53
III.	Upload ảnh .....	54
J.	Tổng kết.....	57

## B. Giới thiệu tác giả

➤ Nguyễn Khánh Hà

- Ngày sinh: 28/7/1991
- Lớp: Công nghệ phần mềm – Khóa 50
- Trường: Đại học Giao thông vận tải Hà Nội

➤ Nguyễn Tiến Dũng

- Ngày sinh: 04/11/1991
- Lớp: Công nghệ phần mềm – Khóa 50
- Trường: Đại học Giao thông vận tải Hà Nội

➤ Vũ Văn Thành Luân

- Ngày sinh: 27/9/1991
- Lớp: Công nghệ phần mềm – Khóa 50
- Trường: Đại học Giao thông vận tải Hà Nội

## C. Mô tả khái quát về RVX

### I. Giới thiệu chung

Công ty cổ phần phần mềm VIAMI mong muốn được mang đến cho các công ty và doanh nghiệp những giải pháp về công nghệ thông tin, hi vọng sẽ giúp doanh nghiệp kiểm soát hiệu quả nguồn lực của mình.

Với kinh nghiệm 15 năm trong việc phát triển giải pháp doanh nghiệp tại thị trường trong và ngoài nước. Sở hữu công nghệ lõi vững chắc và đẩy mạnh phát triển cộng đồng phát triển giải pháp. Có thể cung cấp giải pháp doanh nghiệp tổng thể và tư vấn độc lập.

VIAMI Software với sản phẩm RVX Manager và tiền thân của nó – VIP Enterprise, là một hệ thống giải pháp ERP tiêu chuẩn, có tính mở và linh hoạt cao, có khả năng thích ứng với bất kỳ hình thức kinh doanh nào nhằm giúp các nhà lãnh đạo kiểm soát hiệu quả hoạt động doanh nghiệp mà không phải quá cố gắng về tài chính. VIAMI Software luôn coi khách hàng là người quyết định mọi thành công của chính mình.

### II. Lịch sử hình thành và phát triển

Sáng lập viên, ông Lê Ngọc Quang, một công dân Việt Nam đã từng học tập, sống và làm việc tại Rumani gần 30 năm. Từ 1993, thành lập công ty “VIAMI Investment SRL”. Năm 1995, công ty “VIAMI Computers SRL” khởi đầu các hoạt động trên lĩnh vực công nghệ thông tin, từ kinh doanh phần cứng và lắp ráp máy tính đến xuất nhập khẩu và phân phối linh kiện máy tính. VIAMI Computers đã tích cực đầu tư vào lĩnh vực phần mềm và phát triển sản phẩm quản trị doanh nghiệp mang tên “VIP” từ năm 1997 sau đó đến năm 2000 phát triển sản phẩm mới mang tên “VIP 2000” và tiếp sau là VIP Pro, VIP Enterprise với trọng tâm quản trị kinh doanh dành cho chủ doanh nghiệp. Năm 2003, “VIAMI Software SRL” đã được thành với sản phẩm “VIP Pro” khẳng định cho hướng đi của “VIAMI Software” là phát triển phần mềm ERP dành cho các doanh nghiệp vừa và nhỏ và đã nhanh chóng chiếm lĩnh thị trường Rumani, ([www.viamisoftware.ro](http://www.viamisoftware.ro)). Với những kinh nghiệm và thành quả đã đạt được, ông Lê

Ngọc Quang đã quyết tâm trở về phục vụ đất nước bằng việc thành lập doanh nghiệp mang tên “Công ty Cổ phần Phần mềm VIAMI” (VIAMI Software JSC) tại Việt Nam từ cuối năm 2004, ([www.viamisoftware.com](http://www.viamisoftware.com)). Là nhà phân phối của BitDefender tại Việt Nam, VIAMI Software đã gây tiếng vang mạnh mẽ trên thị trường an toàn dữ liệu trên khắp mọi miền và đã trở thành địa chỉ quen thuộc đáng tin cậy của người dùng Việt Nam. Phương châm của VIAMI Software là: **“LỚN LÊN CÙNG THÀNH CÔNG CỦA KHÁCH HÀNG!”**

### III. Công nghệ sử dụng

#### 1. PHP

Framework giống như 1 thư viện mã lệnh được xây dựng sẵn để chúng ta sử dụng, PHP là 1 ngôn ngữ script rất phổ biến hiện nay bởi những lý do: linh hoạt, dễ sử dụng, dễ học .. nhưng đôi khi việc viết mã PHP, hay bất cứ ngôn ngữ (lập trình) nào khác, có thể trở nên đơn điệu và lủng củng. Vậy nên việc tạo ra PHP frameworks làm cho sự phát triển của những ứng dụng web viết bằng ngôn ngữ PHP trở nên trôi chảy hơn, bằng cách cung cấp 1 cấu trúc cơ bản để xây dựng những ứng dụng đó. PHP framework giúp đỡ các bạn thúc đẩy nhanh chóng quá trình phát triển ứng dụng, tiết kiệm được thời gian, tăng sự ổn định cho ứng dụng.

Trong phần mềm quản trị doanh nghiệp này, PHP framework đã được tạo ra với tên gọi RVX Manager, khi đó ta chỉ cần tạo các file XML để thao tác tạo Module một cách nhanh chóng.

#### 2. Ext Js

Ext Js (Extensal Javascript) là một Framework ứng dụng Javascript thuần túy cho việc xây dựng sự tương tác lẫn nhau giữa các ứng dụng Web có sử dụng công nghệ như Ajax, DHTML và DOM scripting. Đầu tiên nó được xây dựng như 1 thư viện mở rộng từ Yahoo UI bởi Jack Slocum, Ext JS bao gồm sự tương tác giữa jQuery và Prototype.

Bắt đầu với phiên bản 1.1, Ext Js không phụ thuộc vào các thư viện bên ngoài mà thay vào đó, nó có những lựa chọn riêng cho mình.

Ext Js bao gồm tập hợp các “GUI-based form controls” sử dụng cho các ứng dụng Web như: Text field, textarea, daten, [List box](#) and [combo boxes](#), radio, grid vector graphics charts... Có nhiều controls cho phép giao tiếp với một Web server có sử dụng Ajax

### Các phiên bản:

*Ext JS phiên bản 2.0* được phát hành vào ngày 04/12/2007, cung cấp một giao diện và tính năng tương tự như những truyền thống kết hợp với các ứng dụng máy tính để bàn.

*Ext JS phiên bản 3.0* được phát hành vào ngày 06/7/2009, hỗ trợ truyền thông cho REST và Ext.Direct bên nền tảng máy chủ mới.

*Ext JS phiên bản 4.0* được phát hành vào ngày 26/4/2011. Nó bao gồm một cấu trúc lớp sửa đổi hoàn toàn, một sửa đổi dữ liệu gói, một gói phần mềm hoạt hình và bản vẽ sử dụng SVG và VML, và biểu đồ sửa đổi và Theming.

Tham khảo thêm tại: [http://en.wikipedia.org/wiki/Ext\\_JS#cite\\_note-1](http://en.wikipedia.org/wiki/Ext_JS#cite_note-1),  
<http://www.sencha.com/>

## 3. Mysql

**MySQL** là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, ...



**MySQL** là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

**MySQL** được sử dụng cho việc hỗ trợ [PHP](#), [Perl](#), và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,...

#### IV. Khái quát các tính năng và độ ưu việt

RVX Manager là một hệ thống quản trị doanh nghiệp tổng thể nhằm hỗ trợ các cấp lãnh đạo ra quyết định nhanh chóng và chính xác với các tính năng và độ ưu việt sau:

- **Toàn diện:** Giải pháp ERP toàn diện có thể đáp ứng cho bất kỳ tổ chức thương mại hoặc phi thương mại nào.
  - **Tùy biến :** Dễ dàng tùy chỉnh để phù hợp với kiểu dáng, giao diện và báo cáo riêng.
  - **Độ tin cậy :** RVX Manager đem lại sự ổn định, vững chắc và chuẩn xác.
  - **Trực tuyến :** Người dùng có thể truy cập dữ liệu công ty từ xa bằng bất kỳ thiết bị hoặc hệ điều hành nào kể cả điện thoại di động.
  - **Chặt chẽ :** RVX Manager nâng cao hiệu quả và quản lý nhờ tính tập trung và các quy trình thông minh của nó.
  - **Hỗ trợ ra quyết định:** Cung cấp công cụ mạnh mẽ với mô hình phân tích 3 chiều (Khối quyết định) để đánh giá quá trình kinh doanh và ra quyết định đúng đắn.
  - **Giao diện thân thiện:** Giao diện web như desktop là đặc trưng của RVX Manager, có thể dễ dàng điều chỉnh, chạy với mọi hệ điều hành.
  - **Linh hoạt :** Dễ dàng và nhanh chóng thích nghi với mọi điều kiện và hình thức kinh doanh của mỗi tổ chức.
  - **Khả năng mở rộng :** Có thể thay đổi kích thước, chức năng... tùy theo nhu cầu và môi trường kinh doanh mà không ảnh hưởng đến hiệu suất khai thác hàng ngày.
- Tích hợp dữ liệu:** Có thể tích hợp song phương với các hệ thống dữ liệu khác của tổ chức nếu được phép.

## V. Hướng dẫn cài đặt

### 1. Cài đặt trên Windows

- Tải và cài bản *RVX Manager* từ trang

"[http://www.quantridoanhnghiep.biz/?page\\_id=325](http://www.quantridoanhnghiep.biz/?page_id=325)"

- Cài đặt tự động bằng installation kit theo hướng dẫn cài đặt

- Những điểm cần chú ý

- RVX Manager bắt buộc phải được cài mặc định trên thư mục C:\rvx.

RVX Manager chạy trên nền Apache/MySQL.

- Nếu bạn cài luôn Apache – MySQL thì bạn có thể tích chuột vào *Install Apache and MySQL services* ở phần cuối cửa sổ cài đặt. Sau khi cài đặt xong, để kiểm tra dịch vụ Apache-mysq, bạn cần vào <http://localhost/> để kiểm tra lại hoặc dùng phpinfo để kiểm tra ẩn.

- Chú ý bạn hãy cho phép apache.exe và mysql.exe thông qua tường lửa của bạn.

- Cài đặt server Windows khi đã có dịch vụ PHP hoặc cài đặc biệt riêng. Bạn có thể tải bản WAMP (Windows Apache MySQL PHP từ địa chỉ:

(<http://www.wampserver.com/en/#download-wrapper>) và cài đặt trên server.

### 2. Cài đặt trên Linux

- Chuẩn bị trước

- Hãy kiểm tra trên máy (server) các dịch vụ *Apache*, *PHP 5.x* và *MySQL* đã được cài đặt, cấu hình và hoạt động trên servers Linux. Nếu bạn chạy trên máy trạm UBUNTU, bạn có thể chạy lệnh sau đây để cài LAMP một cách nhanh chóng nhất: `sudo apt-get install php5 php5-gd php5-imagick php5-mcrypt php5-mysql php5-sqlite php5-suhosin mysql-server apache2` sau đó bạn hãy kiểm tra bằng phpinfo() xem nó đã chạy chưa

- Tải *IonCube extension* tại: <http://www.ioncube.com/loader.php>

- Cài đặt *IonCube extension* trên server của bạn như được hướng dẫn mà chúng tôi đã tạm dịch lại <http://www.quantridoanhnghiep.biz/huong-dan-cai-dat-ioncube-loader.html> hoặc [http://www.ioncube.com/loader\\_installation.php](http://www.ioncube.com/loader_installation.php)

- Sau đó khởi động lại dịch vụ apache.

### ➤ Cài đặt

- Tải gói *RVX Manager Linux* từ trang <http://www.rvxmanager.com/download-erp/>
- Giải nén (*Unzip*) *rvxmanager.zip* vào thư mục “web root directory”

ví dụ /var/www/html/xxx.

## 3. Sử dụng

Sau khi cài đặt xong, bạn thực hiện các bước sau để sử dụng:

- Bạn mở trình duyệt internet (firefox, chrome, internet explorer,...) và chọn <http://localhost/rvx/>
- Bạn chọn ngôn ngữ sử dụng
- Nếu các dịch vụ đang chạy, bạn sẽ thấy trang *RVX Manager – Tạo cơ sở dữ liệu*
- Company: Nhập tên Công ty của bạn
- MySQL\_Username: Root (không thay đổi, nếu bạn dùng trên máy cá nhân, nếu bạn dùng trên server thì phải tạo 1 người dùng cho CSDL của bạn)
- MySQL\_Password: \_ (để trống theo mặc định, nếu bạn dùng trên máy cá nhân, nếu bạn dùng trên server thì phải tạo 1 mật khẩu cho người dùng cho CSDL của bạn)
- MySQL\_Database: Company (bạn có thể thay đổi, tên không có ký tự đặc biệt, không có dấu và không có khoảng cách)
- Nhấn nút *Tạo cơ sở dữ liệu* và chờ cho đến khi cơ sở dữ liệu được tạo ra, sau đó nhập vào *Finish*
- Bạn sẽ được chuyển sang trang: *RVX Manager – Đăng nhập*
- Company (Công ty): Chọn tên Công ty của bạn
- Username( Tên): admin (mặc định)

- Password (Mật khẩu): admin (mặc định)
- Language (Ngôn ngữ): Việt Nam/English
- Nhấn **Đăng nhập** và bạn sẽ đăng nhập thành công.
- Chúc mừng bạn đã đến với RVX Manager của VIAMI software.

## D. Tổng quan về module

### I. Tổng quan

#### 1. Module

Một module bao gồm các thành phần sau:

- File **menu.xml** (\*): khai báo các model
- Folder, file **controller** của từng model
- Thư mục **\_reports**
- Thư mục **\_dashboard**
- Thư mục **\_locale**

Trong đó, các thành phần (\*) là bắt buộc phải có.

#### 2. Model

Một model bao gồm các thành phần sau:

- Folder **model** cùng tên với **model**
- **List.xml** (\*)
- **Form.xml**
- **Model.xml**
- **Print.xml**
- **Print[i]\_qrp.xml**
- **Print[i]\_sql.xml**
- File **controller** (\*) cùng tên với **model**

Trong đó, các thành phần (\*) là bắt buộc phải có.



### 3. Các file khác

#### ❖ Modules.xml

- Chức năng: khai báo module hiển thị trên RVX
- Danh sách thẻ:

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Thẻ bao các thẻ khác Cú pháp: <xml> </xml>			
<b>module</b> : Khai báo module Cú pháp: < module ... />	id	Text	Tên module
	tittle	Text	Tên hiển thị của module

- Ví dụ:

```
//=====
xml>
  <module id="parent_model" title="Paretn Model" />
  ...
</xml>
//=====
```

## II. Cấu trúc các thành phần

### 1. Module

#### 1.1. Các model và file controller

Mỗi module có thể có 1 hoặc nhiều model.

Mỗi model là 1 mẫu giúp người dùng có thể giao tiếp với cơ sở dữ liệu (xem, tìm kiếm, thêm mới, xóa, sửa,...) một cách trực quan. Bên cạnh đó, model cũng cung cấp khả năng tạo các báo cáo đơn giản (báo cáo trên form).

#### 1.2. Menu.xml

Chức năng: khai báo cách các model hiển thị trên RVX

Danh sách thẻ:

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml:</b> Thẻ bao các thẻ khác Cú pháp: <xml> </xml>			
<b>menu:</b> Khai báo menu Cú pháp: < menu ... />	id	Text	Id của model
	name	Text	Tên model
	folder	True/False	Hiển thị dạng folder hay không
	tittle	Text	Tên hiển thị của model

Ví dụ:

```
//=====
<xml>
  <menu id="1" name="sale1" title="Parent Model" folder="true" />
  <menu id="1.1" name="parent_model" title="Sale Invoices" />
  ...
  <menu id="10" name="reports" title="Reports" />
</xml>
//=====
```

### 1.3. File controller

Trong folder [module], tạo file **\*.php** có tên ứng với model.

Ví dụ:

```
//=====
// File controller parent_model.php có class Parent_Model extends RController
<?php
include(RVXPATH. "controller.php");
class Parent_Model extends RController
{
    function Parent_Model ()
    {
        parent::RController();
        ...
    }
    ...
}
?>
//=====
```

## 1.4. Report

Report là phần chức năng tạo báo cáo thống kê của RVX. Để thực hiện chức năng này, ta cần có các file gồm file \*.xml và file reports.xml để định nghĩa hiển thị báo cáo nằm trong thư mục **\_reports**. Thư mục **\_reports** này cùng với file **report.php** nằm trong thư mục module.

### 1.4.1. Thư mục **\_reports**

- ❖ Chứa các file báo cáo thống kê.
- ❖ Cấu trúc:
  - ⇒ Các file **\*.xml**: các file báo cáo thống kê, tên file là id của báo cáo được sử dụng để gọi và khai báo. Vd: 1001.xml, 2003.xml.
  - ⇒ Danh sách thẻ:

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Thẻ bao các thẻ khác Cú pháp: <xml> </xml>			
<b>title</b> : Khai báo tên báo cáo Cú pháp: <title></title>			
<b>select</b> : Chứa câu lệnh sql cuối Cú pháp: < select ></ select >			
<b>Fields</b> : Chứa các thẻ fields Cú pháp: < fields ></ fields >			
<b>field</b> : Khai báo trường hiển thị Cú pháp: <field ... />	name	Text	Tên trường dữ liệu
	type	Các giá trị: <ul style="list-style-type: none"> <li>▪ String</li> <li>▪ Date</li> <li>▪ Money</li> <li>▪ Integer</li> <li>▪ Number</li> </ul>	Kiểu dữ liệu của cột
	width	Number	Độ rộng của trường

	hidden	True, false	Xác định có hiển thị cột trên report hay không
	caption	Text	Tên tiêu đề của cột
	align	Text	Canh lề cho dữ liệu xuất ra
<b>group::</b> Khai báo trường được group Cú pháp: < group ... />	name	Text	Tên trường dữ liệu
	type	Các giá trị: <ul style="list-style-type: none"> <li>▪ String</li> <li>▪ Date</li> <li>▪ Money</li> <li>▪ Integer</li> <li>▪ Number</li> </ul>	Kiểu dữ liệu của cột
	width	Number	Độ rộng của trường
	caption	Text	Tên tiêu đề của cột
	align	Text	Canh lề cho dữ liệu xuất ra
<b>params:</b> Chứa các thẻ param Cú pháp: < params ></ params >			
<b>param:</b> Khai báo các phần lọc của	control	Các giá trị: <ul style="list-style-type: none"> <li>▪ Lookup</li> <li>▪ Combo</li> <li>▪ Text</li> </ul>	Kiểu dữ liệu của control
	type	Các giá trị: <ul style="list-style-type: none"> <li>▪ String</li> <li>▪ Integer</li> </ul>	Kiểu dữ liệu của trường
	name	Text	Tên của các trường dữ liệu trong bảng
	lookupmodel	Text	Đường dẫn tới các Module
	lookuptable	Text	Tên bảng dữ liệu



báo cáo <i>Cú pháp:</i> < param ... />	lookupname	Text	Tên trường dữ liệu hiển thị của bảng dữ liệu
	lookupkey	Text	Tên trường dữ liệu cần tìm
	comboitems	Text Text ... Text	
	combovalues	Number Number ... Number	
<b><i>initsql:</i></b> Chứa các câu lệnh sql khởi nguồn <i>Cú pháp:</i> < initsql ></ initsql >			

- File **reports.xml**:
  - ❖ Khai báo hiển thị báo cáo trên RVX
  - ❖ Danh sách thẻ:

Thẻ	Thuộc tính	Giá trị	Mô tả
<b><i>xml:</i></b> Thẻ bao các thẻ khác <i>Cú pháp:</i> <xml> </xml>			
<b><i>Folde:::</i></b> Khai báo các thẻ <folder> <i>Cú pháp:</i> < folders ></ folders >			
<b><i>folder:</i></b> Khai báo các folder báo cáo khi hiển thị <i>Cú pháp:</i> < folder .../ >	id	Number	khai báo id cho folder
	name	Text	khai báo tên của folder

<p><b>reports:</b> Khai báo các thẻ &lt;report&gt;</p> <p><b>Cú pháp:</b> &lt; reports &gt;&lt;/ reports &gt;</p>			
<p><b>report:</b> Khai báo các báo cáo và cách hiển thị</p> <p><b>Cú pháp:</b> &lt; report ... /&gt;</p>	id	Number	khai báo id cho báo cáo
	name	Text	khai báo tên của báo cáo
	fd	Text	khai báo folder cho báo cáo

#### 1.4.2. Reports.php

File này nằm trong thư mục module có báo cáo thống kê, đây là file đã được mã hóa.

### 1.5. Dashboard

Dashboard là phần chức năng tạo biểu đồ của RVX. Để có thể thực hiện chức năng này, ta cần các file biểu đồ (\*.xml) trong thư mục **\_dashboard** và file **dashboard.php** nằm trong thư mục module

#### 1.5.1. Thư mục \_dashboard

- ❖ Chứa các file biểu đồ dạng (\*.xml), trong đó \* là id của biểu đồ.

Ví dụ: 1000.xml, 2001.xml,...

- ❖ Danh sách thẻ:

Thẻ	Thuộc tính	Giá trị	Mô tả
<p><b>xml:</b> Thẻ bao các thẻ khác</p> <p><b>Cú pháp:</b> &lt;xml&gt; &lt;/xml&gt;</p>			
<p><b>Title:</b> báo tên báo cáo</p> <p><b>Cú pháp:</b> &lt;title&gt;&lt;/title&gt;</p>			

<b><i>select</i></b> : Chứa câu lệnh sql cuối <b>Cú pháp</b> : < select ></ select >			
<b><i>char</i></b> : Khai báo các dạng đồ thị và kiểu hiển thị dữ liệu <b>Cú pháp</b> : <char ... />	type	Các giá trị: <ul style="list-style-type: none"> <li>▪ Pie</li> <li>▪ Column</li> <li>▪ Line</li> </ul>	Đồ thị dạng Quạt, Cột, Đường
	width	Number	Độ rộng của đồ thị
	colspan	Number	Màn hình mặc định được chia làm 3 cột dạng ẩn, <b>colspan</b> chính là số cột cấp cho đồ thị.
	legend	True, False	Liệt kê và giải thích các màu, biểu tượng, ký hiệu và các chú giải có trên bản đồ khi có giá trị True. (Áp dụng cho biểu đồ tròn)
<b><i>initsql</i></b> : Chứa các câu lệnh sql khởi nguồn <b>Cú pháp</b> : < initsql ></ initsql >			

### 1.52. Dashboard.php

File này nằm trong thư mục module có biểu đồ, đây là file đã được mã hóa.

### 1.6. Locale

Thư mục có chứa các file chuyển đổi ngôn ngữ như *sale\_en.php*, *sale\_ro.php*, *sale\_vn.php*,..

## 2. Model

### 2.1. Các file cần thiết trong model

#### 2.1.1. List.xml

##### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Chứa cấu trúc để hiển thị list. <i>Cú pháp</i> : <xml> </xml>			
<b>view</b> : Khai báo hiển thị list trên RVX. <i>Cú pháp</i> : <view ... />	title	Text	Tên hiển thị của list
	formwidth	Number	Độ rộng của list
	formheight	Number	Độ cao của list
<b>sql</b> : Chứa câu lệnh SQL để lấy dữ liệu từ CSDL. <i>Cú pháp</i> : <sql></sql>			
<b>columns</b> : Chứa các thẻ column khai báo các cột cho list <i>Cú pháp</i> : <columns></ columns>			
<b>column</b> : Khai báo cột cho list <i>Cú pháp</i> : < column .../>	Field	Text	Khai báo cột cho list, giá trị ứng với tên cột trong bảng kết quả từ SQL.
	Type	Các giá trị: <ul style="list-style-type: none"> <li>• String</li> <li>• Bool</li> <li>• Integer</li> <li>• Number</li> <li>• Date</li> <li>• Money</li> </ul>	Kiểu dữ liệu cho cột.
	Hidden	Các giá trị <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	Xác định xem cột có được hiển thị trên list không.
	Caption	Text	Tên cột hiển thị trên list.
	Width	Number	Xác định độ rộng của cột khi hiển thị trên list.
	Align	Các giá trị: <ul style="list-style-type: none"> <li>• Left</li> <li>• Center</li> <li>• Right</li> </ul>	Căn lề cho cột.

## b. Ví dụ

```
//=====
<xml>
  <view title="Parent Model" formwidth="800" formheight="600" />
  <sql>
    SELECT P.Code, P.Name, P.Birthday, P.Sex
    FROM ParentModel P
    ORDER BY P.Name
  </sql>
  <columns>
    <column field="Code" type="string" width="90" />
    ...
  </columns>
</xml>
//=====
```

### 2.1.2. Form.xml

#### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Chứa cấu trúc để hiển thị form. Cú pháp: <xml> </xml>			
<b>form</b> : Khai báo hiển thị list trên RVX. Cú pháp: <form ... />	title	Text	Tên hiển thị của list
<b>boxes</b> : Chứa khai báo các thẻ box. Cú pháp: <boxes> <boxes>			
<b>box</b> : Tạo box cho form. Cú pháp: <box ... />	Id	Number	Định danh phân biệt với các box khác.
	Caption	Text	Tên hiển thị của box.
	Gridmodel	Text	Đường dẫn tới file model để tạo box hiển thị model đó. Đường dẫn này có giá trị tương đối so với thư mục <b>app</b> .
	Width	Number	Độ rộng của box

<p><b>controls:</b> Chứa khai báo các thẻ control.  <b>Cú pháp:</b>                  &lt;controls&gt; &lt;controls&gt;</p>	Không có	Không có	Không có
<p><b>control:</b> Khai báo control trên form  <b>Cú pháp:</b> &lt;control .../&gt;</p>	Box	Number	Xác định xem control thuộc box nào.
	Field	Text	Tên trường ứng với tên cột trong bảng CSDL.
	Type	Các giá trị: <ul style="list-style-type: none"> <li>• Text</li> <li>• Memo</li> <li>• Check</li> <li>• Number</li> <li>• Date</li> <li>• Select</li> <li>• Combo</li> <li>• Radio</li> </ul>	Kiểu dữ liệu của control
	Select	Có cú pháp: Text Text Text  ...	Các giá trị hiển thị cho control nếu type=radio
	Caption	Text	Nhãn của control.
	Width	Number	Độ rộng của control.
	Height	Number	Độ cao của control.
	Onvalidate	Các giá trị: <ul style="list-style-type: none"> <li>• True hoặc 1</li> <li>• False hoặc 0</li> </ul>	Xác định xem control có được kiểm tra khi bị thay đổi dữ liệu hay không.
	Hidden	Các giá trị: <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	Xác định có hiển thị control trên form hay không.

<b>actions:</b> Chứa khai báo các thẻ action. <b>Cú pháp:</b> < actions >< actions >	Không có	Không có	Không có
<b>action:</b> Khai báo action trên form. <b>Cú pháp:</b> <action .../>	Caption	Text	Tên hiển thị của action.
	Url	Text	Đường dẫn tới hàm xử lý action
	Popup	Các giá trị <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	Xác định action có chạy chế độ popup hay không

### b) Ví dụ

```

//=====
<xml>
  <form title="Sale Invoice" />
  <tabs>
    <tab id="1" caption="Document" columns="4" />
    ...
  </tabs>
  <boxes>
    <box id="1" tab="1" caption="Document" width="250" />
    ...
  </boxes>
  <controls>
    <control box="1" field="Id" type="hidden" />
    ...
  </controls>
  <actions>
    <action caption="Price discount"
      url="sale/sale_invoice/action_discount" />
    ...
  </actions>
</xml>
//=====

```

### 2.1.3. Model.xml

#### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Chứa cấu trúc để hiển thị model. <i>Cú pháp</i> : <xml> </xml>			
<b>model</b> : Khai báo model cho module <i>Cú pháp</i> : <model .../>	Table name	Text	Tên bảng trên CSDL
<b>fiels</b> : Chứa khai báo các thẻ field. <i>Cú pháp</i> : : <fields> <fields>			
<b>field</b> : Tạo box cho form. <i>Cú pháp</i> : <field ... />	Name	Text	Tên trường ứng với cột trong bảng CSDL.
	Type	Các giá trị: <ul style="list-style-type: none"> <li>• String</li> <li>• Bool</li> <li>• Integer</li> <li>• Number</li> <li>• Date</li> </ul>	Kiểu dữ liệu của trường.
	Select	Các giá trị: Text Text Text ...	Khi báo các giá trị cho trường khi type của control tương ứng trên model là select.
	Unique	Các giá trị: <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	Xác định xem giá trị của trường có phải là duy nhất hay không.
	Upper	Các giá trị: <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>	Xác định xem giá trị của trường có được tự động đổi sang chữ in hoa hay không.
<b>Relations</b> Chứa khai báo các thẻ relation. <i>Cú pháp</i> : < relations < relations>	Không có	Không có	Không có



<b>relation</b> Khai báo quan hệ với bảng khác. <i>Cú pháp:</i> < relation ... />	Table	Text	Tên bảng quan hệ với bảng của model theo: khóa ngoại – khóa chính
	Key	Text	Tên khóa ngoại trong bảng quan hệ

### b) Ví dụ

```

//=====
<xml>
  <model tablename="HoSoMuon" />
  <fields>
    <field name="Id" />
    <field name="MaHoSoMuon" type="string" unique="true" upper="true"
      mandatory="true" />
    ...
  </fields>
  <relations>
    <relation table="DanhMucSach" key="MaHoSoMuon" />
  </relations>
</xml>
//=====

```

## 2.2 Các file báo cáo trên form

### 2.2.1 Print.xml

#### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Chứa cấu trúc để hiển thị prints. <i>Cú pháp:</i> <xml> </xml>			
<b>prints</b> : chứa các thẻ print con. <i>Cú pháp:</i> <print .../>	Id	Number	Id của các báo cáo
	Caption	Text	Tên hiển thị của báo cáo

### b) Ví dụ

```

//=====
<xml>
  <prints>
    <print id="1" caption="Phiếu bảo hành" />
    <print id="2" caption="Phiếu dịch vụ sửa chữa" />
  </prints>
</xml>
//=====

```

#### 2.2.2 Print\*\_grp.xml

### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>xml</b> : Chứa cấu trúc để hiển thị thẻ con bên dưới đây. <i>Cú pháp</i> : <xml> </xml>			
<b>Reports</b> định nghĩa khổ giấy của Báo cáo (A0, A1...) , mặc định của Báo cáo là khổ A4. <i>Cú pháp</i> : <report .../>	Id	Number	Id của các báo cáo
	Caption	Text	Tên hiển thị của báo cáo
<b>bands</b> : Chứa các thẻ band con. <i>Cú pháp</i> : <bands</bands>			
<b>band</b> : Khai báo độ cao của các thành phần trong báo cáo <i>Cú pháp</i> : <band .../>	name	Text	Tên các thành phần như: header, title, detail, footer, summary.
	height	Number	Độ cao của các thành phần.
<b>ctrls</b> : Chứa các thẻ ctrl con. <i>Cú pháp</i> : < ctrls> </ ctrls>			
	band	Text	Tên các thành phần như: header, title, detail, footer, summary.

<b>ctrl:</b> Tạo khung, tạo nhãn và nội dung của báo cáo. <b>Cú pháp:</b> <ctrl .../>	x, y, w, h	Number (pixel)	x: Tọa độ tính từ bên trái y: Tọa độ tính từ phía trên w: Độ rộng của thành phần h: Độ cao của thành phần
	caption	Text	Nhãn hiển thị trong báo cáo
	align	L, C, R	Căn chữ theo trái, giữa, phải
	fontstyle	B, L, U	Kiểu chữ hiển thị đậm, nghiêng, gạch dưới...
	fontname	Helvetica, arial ...	Tên font chữ
	fontsize	Number	Cỡ chữ
	ftype	String, integer, money	Kiểu dữ liệu của trường
	field	Text	Trường dữ liệu sẽ hiển thị trong báo cáo
	dataset	Number	Các Dataset tương ứng trong file <i>printi_sql.xml</i>
	type	Shape, image	Shape : Tạo khung hình chữ nhật bao các thành phần Image: File Ảnh
url	Path	Đường dẫn tới file ảnh	

### b) Ví dụ

```
//=====
<xml>
<report />

<bands>
  <band name="header" height="95"/>
  <band name="title" height="0"/>
  <band name="detail" height="115"/>
  <band name="footer" height="100"/>
</bands>
```

```

</bands>

<ctrls>

<!-- Logo and information of Company -->
<ctrl band="header" x="0" y="0" w="185" h="43" type="shape" />
  <!-- ..... -->
  <ctrl band="header" x="01" y="03" w="43" h="18" url="[LOGO]" type="image" />
  <!-- ..... -->
  <ctrl band="header" x="50" y="12" caption="Trụ sở:" />
  ...
  <!-- ..... -->
  <ctrl band="header" x="0" y="3" w="185" field="Name" ftype="string"
fontsize="18" align="C" dataset="2" />
  ...
  <!-- ..... -->
  <!-- Title of document ..... -->
  <ctrl band="header" x="0" y="43" w="185" h="17" type="shape" />
  <!-- ..... -->
  <ctrl band="header" x="0" y="45" w="185" caption="HÓA ĐƠN BÁN HÀNG"
fontstyle="B" fontsize="18" align="C" />
  <!-- ..... -->
  <!-- Information of document ..... -->
  <ctrl band="header" x="0" y="60" w="185" h="25" type="shape" />

  <!-- ..... -->
  <ctrl band="header" x="2" y="63" caption="Khách hàng:" />
  ...
  <!-- ..... -->
  <ctrl band="header" x="25" y="63" w="90" field="PartnerName" ftype="string"
dataset="1" align="L" />
  ...
  <!-- ..... -->
  <!-- Header of document ..... -->
  <ctrl band="header" x="0" y="85" w="10" h="10" type="shape" />
  ...
  <!-- ..... -->
  <ctrl band="header" x="0" y="87" w="10" caption="STT" align="C" />
  ...
  <!-- ..... -->
  <!-- Column of document ..... -->
  <ctrl band="header" x="0" y="95" w="10" h="90" type="shape" />
  ...
  <!-- ..... -->
  <ctrl band="detail" x="0" y="2" w="10" h="6" type="counter" align="C" />
  ...
  <!-- ..... -->
  <!-- Summary of document ..... -->
  <ctrl band="header" x="0" y="185" w="185" h="20" type="shape" />

```

```

        <!-- ..... -->
        <ctrl band="footer" x="120" y="6" align="R" caption="Tổng giá trị:" />
        <!-- ..... -->
        <ctrl band="footer" x="133" y="6" w="50" field="ValueTotV" ftype="money"
align="R" dataset="1" />
        <!-- ..... -->
        <!-- .....Ghi chú..... -->
        <ctrl band="header" x="0" y="205" w="185" h="10" type="shape"/>
        <ctrl band="footer" x="3" y="21" caption="Ghi chú:"/>
        <ctrl band="footer" x="20" y="21" w="165" field="Notes" ftype="string" dataset="1"
align="L"/>
        <!-- Footer of document ..... -->
        <ctrl band="header" x="0" y="230" w="185" h="35" type="shape" />
        <!-- ..... -->
        <ctrl band="footer" x="0" y="46" w="40" caption="Khách hàng" align="C" />
        <ctrl band="footer" x="0" y="50" w="40" caption="(chữ ký, họ tên)" align="C" />
        <!-- ..... -->
        <ctrl band="footer" x="90" y="46" w="55" caption="Người lập" align="C" />
        <ctrl band="footer" x="90" y="50" w="55" caption="(chữ ký, họ tên)" align="C" />
        <!-- ..... -->
    </ctrls>
</xml>
//=====

```

### 2.2.3 Print\*\_sql.xml

#### a) Danh sách thẻ

Thẻ	Thuộc tính	Giá trị	Mô tả
<b>Xml:</b> Chứa các thẻ con bên dưới đây. Cú pháp: <xml> </xml>			
<b>datasets:</b> chứa 1 hay nhiều thẻ <dataset></dataset> con Cú pháp: <datasets> </datasets>			
<b>dataset</b> Chứa các câu lệnh truy vấn Sql Cú pháp: <dataset></dataset>			

## b) Ví dụ

```
//=====
<xml>
  <databases>
    <dataset>
      <!-- Các lệnh sql -->
    </dataset>
    ...
  </databases>
</xml>
//=====
```

## E. Cách tạo 1 module đơn giản

### I. Tạo và định nghĩa module mới

#### Các bước thực hiện

- Bước 1: Tạo folder mới trong folder app có tên là tên module cần tạo, chú ý tên module viết thường không dấu.

Ví dụ: *catalog*, *buy* ...

- Bước 2: Mở file modules.xml nằm ở thư mục rvx, khai báo module bằng thẻ:  
**<module id="ten\_module" title="ten\_hien\_thi" />**

Ví dụ:

```
//=====
xml>
  <module id="parent_model" title="Parent Model" />
  <module id="library" title="Quản lí thư viện " />
  <module id="admin" title="Administration" />
  ...
</xml>
//=====
```

- Bước 3: Trong module vừa tạo, tạo các model, \_reports, dashboard, file menu.xml để hiển thị các model trên RVX. Cách tạo model, \_reports, dashboard sẽ được hướng dẫn phía dưới.

## II. Tạo và định nghĩa model mới

### Các bước thực hiện

- Bước 1: Tạo các bảng cơ sở dữ liệu cho module bằng lệnh sql. Lệnh được chạy trong sql command của RVX. Phần tô đậm trong ví dụ dưới đây là bắt buộc phải có trong khi tạo mới một bảng.

Ví dụ:

```
//=====
CREATE TABLE ParentModel(
    Id INTEGER NOT NULL AUTO_INCREMENT,
    Code VARCHAR(32),
    Name VARCHAR(128),
    Birthday DATE,
    Sex SMALLINT,
    Age INTEGER,
    Weight NUMERIC(5,2),
    Description BLOB,
    CreateTime TIMESTAMP,
    UpdateTime TIMESTAMP,
    Note BLOB,
    CreateUser VARCHAR(64),
    UpdateUser VARCHAR(64),

    PRIMARY KEY (Id),
    UNIQUE KEY Code(Code)
) ENGINE=INNODB;
//=====
```

- Bước 2 : Tạo folder mới trong thư mục module có tên là tên model. Chú ý là đặt tên chữ thường, không dấu, có thể phân biệt bởi dấu ‘\_’. Suy ra ta có model tên là ***parent\_model***.
- Bước 3: Tạo 1 file ***php*** có cùng tên với model cần tạo. Trong file ***php*** này, khai báo class extends Rcontroller. Chú ý đặt tên lớp ứng với tên model, viết hoa chữ cái đầu, dùng ‘\_’ để phân cách các từ.

Ví dụ:

```
//=====
//file parent_model.php
<?php
include(RVXPATH. "controller.php");
class Parent_Model extends RController
{
    function Parent_Model ()
    {
        parent::RController();
    }
}
?>
//=====
```

- Bước 4: Tạo folder có tên là **parent\_model**, trong folder này lần lượt tạo 3 file **form.xml**, **list.xml**, **model.xml**.
  - File **form.xml**: định nghĩa giao diện nhập liệu của model.

Ví dụ:

```
//=====
<xml>
<form title="Parent Model" />

<boxes>
    <box id="1" caption="Parent Model" width="400" columns="1" />
</boxes>

<controls>
    <control box="1" field="Id" type="hidden" />
    <control box="1" field="Code" type="text" />
    <control box="1" field="Name" type="text" />
    <control box="1" field="Birthday" type="date" />
    <control box="1" field="Sex" type="check" />
    <control box="1" field="Age" type="integer" />
    <control box="1" field="Weight" type="number" />
    <control box="1" field="Description" type="text" />
</controls>
</xml>
//=====
```



- File **list.xml**: Định nghĩa danh sách hiển thị của model.

Ví dụ:

```
//=====
<xml>
  <view title="Parent Model" formwidth="800" formheight="600" />
  <sql>
    SELECT P.Code, P.Name, P.Birthday, P.Age, P.Sex, P. Weight ,
    P.Description
    FROM ParentModel P
    ORDER BY P.Name
  </sql>

  <columns>
    <column field="Code" type="string" width="150" />
    <column field="Name" type="string" width="150" />
    <column field="Birthday" type="date" width="150" />
    <column field="Age" type="integer" width="150" />
    <column field="Sex" type="bool" width="100" />
    <column field="Weight" type="money" width="100" />
    <column field="Description" type="string" width="200" />
  </columns>
</xml>
//=====
```

- File **model.xml**: Định nghĩa các trường dữ liệu của model.

Ví dụ:

```
//=====
<xml>
  <model tablename="ParentModel" />
  <fields>
    <field name="Id" />
    <field name="Code" type="string" mandatory="true" unique="true"
    upper="true" />
    <field name="Name" type="string" mandatory="true" />
    <field name="Birthday" type="date" mandatory="true" />
    <field name="Age" type="string" mandatory="true" />
    <field name="Sex" type="bool" mandatory="true" />
    <field name="Weight" type="number " mandatory="true" />
    <field name="Description" type="string" mandatory="true" />
  </fields>
</xml>
//=====
```

Bước 5: File *menu.xml* để hiển thị các model trên RVX.

Ví dụ:

```
//=====
<xml>
  <menu id="1" title="Planning" folder="true" />
  <menu id="1.2" name="plan" title="Plan List" />
  ...
  <!--Đây là thẻ khai báo module vừa tạo-->
  <menu id="1.3" name="parent_model " title="Parent Model" />
  ...
  <menu id="2" name="reports" title="Reports" />
</xml>
//=====
```

### III. Tạo report trên form

#### Các bước thực hiện

Trong thư mục của model cần báo cáo, tạo file *print.xml* nếu chưa có, các file *printi\_qrp.xml*, *printi\_sql.xml* với *i* là Id của báo cáo.

Tạo file *print.xml*: Tạo file *print.xml* khai báo các báo cáo đã tạo để hiển thị chúng. Khai báo thẻ `<print id=[id của báo cáo] caption=[tên báo cáo] />` (*id* tương ứng với *i* trong *printi\_qrp.xml*, *printi\_sql.xml*).

Ví dụ:

```
//=====
<xml>
  <prints>
    <print id="1" caption="Print Parent Model" />
    ...
  </prints>
</xml>
//=====
```

File *print1\_sql.xml*: Định nghĩa dữ liệu của chứng từ.

Ví dụ:

```
//=====
<!--file print1_sql.xml -->
<xml>
  <datasets>
```

```

        <dataset>
            SELECT Code, Name, Birthday, Age, Sex
            FROM ParentModel
            WHERE GrandParentsId = :Id
            <!-- Id để lấy bản ghi hiện tại -->
        </dataset>
    </datasets>
</xml>
//=====

```

File *print1\_qrp.xml*: Định nghĩa hiển thị dữ liệu của chứng từ.

```

//=====
<xml>
<report />

<bands>
    <band name="header" height="95"/>
    <band name="title" height="0"/>
    <band name="detail" height="115"/>
    <band name="footer" height="100"/>
</bands>

<ctrls>

<!-- Logo and information -->
<ctrl band="header" x="0" y="0" w="185" h="43" type="shape" />
    <!-- ..... -->
<ctrl band="header" x="01" y="03" w="43" h="18" url="[LOGO]" type="image" />
    <!-- ..... -->
<ctrl band="header" x="50" y="12" caption="Trụ sở:" />
...
<!-- Title of document ..... -->
<ctrl band="header" x="0" y="43" w="185" h="17" type="shape" />
    <!-- ..... -->
<ctrl band="header" x="0" y="45" w="185" caption="Print Parent Model"
fontstyle="B" fontsize="18" align="C" />
<!-- ..... -->
<!-- Information of document ..... -->
<ctrl band="header" x="0" y="60" w="185" h="25" type="shape" />

    <!-- ..... -->
<ctrl band="header" x="2" y="63" caption="Parent:" />
...
    <!-- ..... -->
<ctrl band="header" x="25" y="63" w="90" field="Name" ftype="string"
dataset="1" align="L" />
...

```

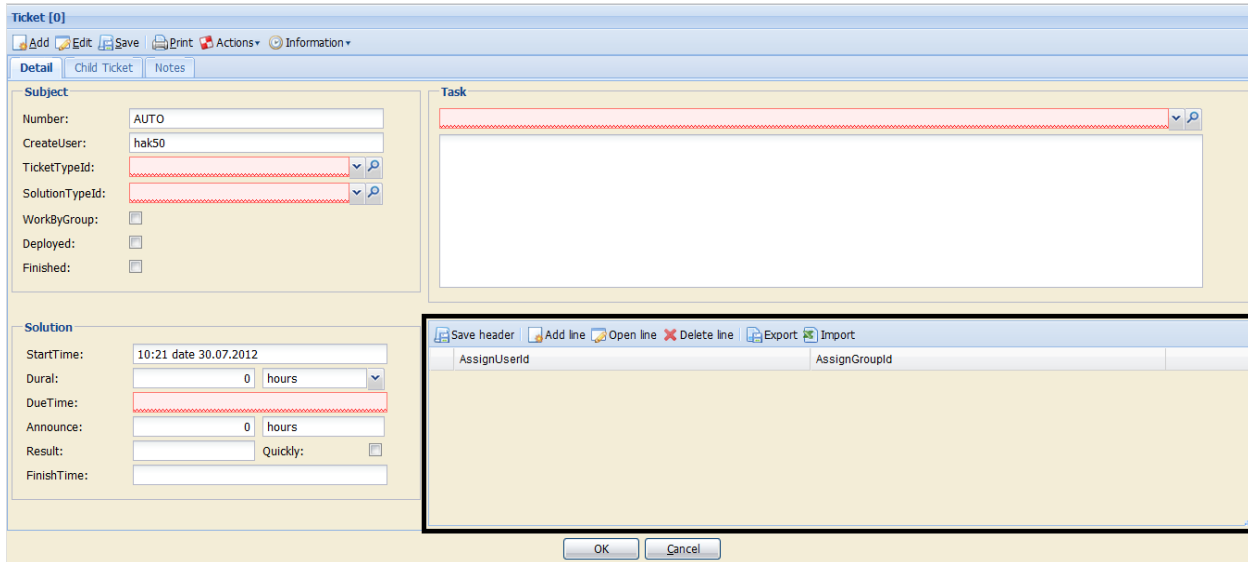
```
<!-- ..... -->
<!-- Header of document ..... -->
<ctrl band="header" x="0" y="85" w="10" h="10" type="shape" />
...
<!-- ..... -->
<ctrl band="header" x="0" y="87" w="10" caption="STT" align="C" />
...
<!-- ..... -->
<!-- Column of document ..... -->
<ctrl band="header" x="0" y="95" w="10" h="90" type="shape" />
...
<!-- ..... -->
<ctrl band="detail" x="0" y="2" w="10" h="6" type="counter" align="C" />
...
<!-- ..... -->
<!-- Summary of document ..... -->
<ctrl band="header" x="0" y="185" w="185" h="20" type="shape" />

<!-- .....Ghi chú..... -->
<ctrl band="header" x="0" y="205" w="185" h="10" type="shape"/>
<ctrl band="footer" x="3" y="21" caption="Ghi chú:" />

<!-- Footer of document ..... -->
<ctrl band="header" x="0" y="230" w="185" h="35" type="shape" />
<!-- ..... -->
<ctrl band="footer" x="90" y="46" w="55" caption="Người lập" align="C" />
<ctrl band="footer" x="90" y="50" w="55" caption="(chữ ký, họ tên)" align="C" />
<!-- ..... -->
</ctrls>
</xml>
//=====
```

## F. Tạo Model Cha, Con

Rvx hỗ trợ việc hiển thị list của một model trong form của một model khác.



### I. Liên kết khóa ngoại

Tạo 1 model mới có tên `child_model`, khi nhập dữ liệu cho bảng con thì ta sẽ tham chiếu tới bảng cha để lấy dữ liệu qua combobox, đó chính là dữ liệu được lấy từ model `parent_model` vừa tạo phía trên.

#### Các bước thực hiện

Bước 1: Tạo cơ sở dữ liệu, tương tự `ParentModel`, bổ xung thêm trường `ParentModelId INTEGER (Cú pháp [Tên bảng]+ hậu tố Id)` làm khóa ngoại

Bước 2: Tạo 1 file controller ***php*** cho model tương tự model ***parent\_model***.

- Bước 3: Tiếp tục tạo 1 folder có tên là ***child\_model***, trong folder này lần lượt tạo 3 file ***form.xml***, ***list.xml***, ***model.xml*** giống với model ***parent\_model***
  - File ***form.xml***: bổ xung thêm .thẻ sau để định nghĩa khóa ngoại

```
<control box="1" field="ParentModelId" type="combo" />
```

- File **list.xml**: bổ xung thêm như sau:

Ví dụ:

```
//=====
...
<sql>
    SELECT C.Id, C.Code, C.Name, P.Name AS ParentName,
    FROM ChildModel C, ParentModel P
    WHERE C.ParentModelId = P.Id
    ORDER BY C.Name
</sql>

<columns>
    <!--Định nghĩa khóa ngoại -->
    <column field=" ParentName " type="string" width="150" />
    ...
</columns>
//=====
```

- File **model.xml**: bổ xung thẻ sau để định nghĩa dữ liệu tham chiếu tới Combobox. `<field name=" ParentModelId " type="integer" mandatory="false" lookupmodel="family/parent_model" lookuptable="ParentModel" lookupname="Name"/>`

## II. Tạo form Cha, Con

Ứng với mỗi **parent**, ta sẽ hiển thị một List danh sách các **child** của **parent** đó có.

### Các bước thực hiện

Bước 1: Tạo CSDL có tên ParentModelLine để model parent\_model tham chiếu tới. tương tự ParentModel, bổ xung trường sau làm khóa ngoại  
*"ParentId INTEGER"*

- Bước 2: Tạo 1 file controller **php** như tạo model đơn gian, thêm 1 class mới kế thừa từ class **RModel** với phương thức **Insert()** để tự động thêm Id của ParentModel vào trường ParentId của ParentModelLine.

Ví dụ:

```
//=====
class Parent_Model_Line_Model extends RModel
```

```

{
    // Tự chèn Id của bảng cha vào ParentId của bảng con tương ứng
    public function Insert()
    {
        parent::Insert();
        $rvx = & get_engine();
        $f['ParentsId'] = $rvx->Context->getParam('parentid');
        $this->setFields($f);
    }
}
//=====

```

- Bước 3: Tạo folder có tên là **parent\_model\_line**, trong folder này lần lượt tạo 3 file **form.xml**, **list.xml**, **model.xml**. Tương tự như tạo 1 model đơn giản.
- Bước 4: Chỉnh sửa 2 file **form.xml**, **model.xml** của model **parent\_model**
  - File **form.xml**: thêm 1 thẻ <box...> để đi tới model con

Ví dụ:

```

//=====
...
<!--Thẻ được thêm mới-->
<box id="2" height="200" width="700"
...
//=====

```

- File **model.xml**: Định nghĩa các trường dữ liệu của model.

Ví dụ:

```

//=====
...
<!--Thẻ được thêm mới sẽ liên kết với bảng con qua khóa ngoại-->
<relations>
    <relation table="ChildModel" key="ParentsId" />
</relations>
...
//=====

```

## G. Tạo module nâng cao

### I. File Controller

Cách gọi hàm để sử dụng:

```
//=====
function function_name()
//=====
{
    $rvx =& get_engine();
    parent::function_name ();
    ...
}
```

#### 1. Lớp Controller

##### a) Thông tin:

Lớp cha: RController.

##### b) Phương thức:

	Phương thức	Param	Mô tả
1	RController		Hàm định nghĩa Controller
2	CheckRight	( \$CanAccess )	Kiểm tra đúng (CanAccess)
3	CreateList		Tạo List
4	CreateModel		Tạo Model
5	CreateForm		Tạo Form
6	Index		Chỉ mục
7	Fetch		Tìm nạp
8	Add		Thêm



<b>9</b>	View		Xem
<b>10</b>	ViewLink		Xem đường dẫn
<b>11</b>	Edit		Sửa
<b>12</b>	CheckEdit		Kiểm tra sửa
<b>13</b>	Save		Lưu
<b>14</b>	Delete		Xóa
<b>15</b>	Select		Chọn
<b>16</b>	Combo		Combo
<b>17</b>	CtrlValidate		Kiểm tra tính hợp lệ
<b>18</b>	Printer		Máy in
<b>19</b>	Audit		Kiểm nghiệm
<b>20</b>	CheckPosted		Kiểm tra Post
<b>21</b>	SetPosted		Đặt Post
<b>22</b>	Export_Excel		Xuất Excel
<b>23</b>	Import_Excel		Nhập Excel
<b>24</b>	RedirectError		Chuyển tiếp lỗi
<b>25</b>	InfoNote		Ghi chép thông tin
<b>26</b>	Inspector		Thanh tra

## 2. Lớp Model

### a) Thông tin:

Lớp cha: RModel

### b) Phương thức:

	Phương thức	Param	Mô tả
1	RModel		Hàm định nghĩa Model
2	SetController	( \$ctrl )	Đặt Controller
3	AddField	( \$field, \$value )	Thêm trường
4	Load		Tải
5	LoadXml	( \$url )	Tải Xml
6	Open	( \$id )	Mở
7	Insert		Thêm
8	Edit	( \$id )	Sửa
9	Post	( \$id )	Đăng
10	Save	( \$id )	Lưu thay đổi
11	SaveInsert	( \$id )	Lưu thêm mới
12	SaveUpdate	( \$id )	Lưu thay đổi
13	Delete	( \$id )	Xóa
14	DeleteChildren	( \$id )	Xóa con

<b>15</b>	RenderCombo		Hoàn trả Combo
<b>16</b>	Validate		Phê duyệt
<b>17</b>	SetField	( \$field, \$value )	Đặt trường
<b>18</b>	GetField	( \$id )	Lấy trường
<b>19</b>	SetFields	( \$fields, \$values )	Đặt các trường
<b>20</b>	GetFields		Lấy các trường
<b>21</b>	SaveField		Lưu trường
<b>22</b>	CheckRelations		Kiểm tra các mối quan hệ
<b>23</b>	CheckPosted		Kiểm tra Post
<b>24</b>	AddAuditFields		Thêm trường kiểm nghiệm
<b>25</b>	RenderAudit		Hoàn trả kiểm nghiệm
<b>26</b>	ValidateDate	( \$id )	Phê duyệt ngày
<b>27</b>	OnBeforeSave	( \$id )	Xử lý trước khi lưu
<b>28</b>	OnAfterSav	( \$id )	Xử lý trước khi lưu
<b>29</b>	OnBeforeDelete	( \$id )	Trước khi xóa
<b>30</b>	OnAfterDelete	( \$id )	Sau khi xóa
<b>31</b>	CheckFieldsChanged		Kiểm tra các trường thay đổi
<b>32</b>	GenerateNumber	( \$field )	Tạo số
<b>33</b>	GetDescriptionString		Lấy chuỗi mô tả

### 3. Lớp List

#### a) Thông tin:

Lớp cha: Rlist

#### b) Phương thức:

	Phương thức	Param	Mô tả
1	RList		Hàm định nghĩa List
2	SetController	( \$ctrl )	Đặt Controller
3	Load		Tải
4	LoadXml	( \$url )	Tải XML
5	GetModelUrl		Lấy Url của model
6	ExtJsRecord		Bản ghi ExtJs
7	ExtJsModel		Model ExtJs
8	ExtJsSearchColumns		Tìm các cột ExtJs
9	ExtJsFirstColumn		Cột đầu tiên ExtJs
10	ExtJsFilters		Lọc ExtJs
11	FetchData		Tìm nạp dữ liệu
12	FilterSql	( \$sql )	Lọc sql
13	FetchDataSend		Tìm nạp dữ liệu gửi
14	GetColumnIndex		Lấy chỉ mục của cột
15	Render		Hoàn trả
16	RenderActions		Hoàn trả hành động
17	RenderSelect		Hoàn trả chọn
18	RenderParams		Hoàn trả tham số

## 4. Lớp Form

### a) Thông tin:

Lớp cha: RForm

### b) Phương thức:

	Phương thức		Mô tả
1	Load		Nạp
2	LoadXml	( \$url )	Nạp Xml
3	Render		Hoàn trả
4	RenderControl	( \$ctrl )	Hoàn trả control
5	RenderPage		Hoàn trả trang
6	SetModel		Đặt model
7	RenderButtons		Hoàn trả các nút

## II. Lọc List

### Các bước thực hiện

Bước 1: Trong file .php cần thao tác, thêm *function RenderControl( \$ctrl )*, *function RenderParents()* vào lớp kế thừa từ lớp *Rform*.

Bước 2: Trong *function RenderControl( \$ctrl )* sẽ có tham số vào là chuỗi html chưa được lọc, kết quả trả về là chuỗi html sau khi lọc khi lọc. Kiểm tra nếu *\$ctrl->Name == "Id của trường xảy ra sự kiện lọc"* là đúng thì sẽ thực hiện *function RenderParents()*.

Ví dụ:

```
//=====
function RenderControl( $ctrl )
{
    $html = parent::RenderControl( $ctrl );

    // filter parent by selected grandparent
    if( $ctrl->Name == 'ParentsId' )
    {
        $html.= $this->RenderParents();
    }
    return $html;
}
//=====
```

Bước 3: Tạo **function RenderParents()**: function sẽ trả về các bản ghi có Name của bảng cùng tên với **Name** trong Combobox cha mà ta cần lọc. sự kiện được gọi khi người dùng chọn vào biểu tượng hình “Kính lúp” ngay cạnh Combobox *Id của trường xảy ra sự kiện lọc*

```
//=====
function RenderParents()
{
    return
    "
    try
    {
        var grand_name = null;
        ParentsId.onTrigger2Click= function() {
            grand_name = Ext.getCmp('GrandparentsId').getRawValue();
            HandleTrigger('http://localhost/rvx/index.php?family/parents/index/grand_name/' + grand_name);
            ParentsIdStore.baseParams = {lookupkey:'Id', lookupname:'Name', grand_name:grand_name};
            ParentsIdStore.load();
        };
    }
}
//=====
```

Bước 4: Trong file .php của model có dữ liệu được lọc, thêm **class Parents\_List** **extends Rlist** có chứa function **function FilterSql( \$sql )** để lọc câu lệnh sql. Hàm này sẽ lọc theo dữ liệu vào ứng với trường Name, kết quả trả về là câu lệnh sql chọn các bản ghi đã được lọc.

Ví dụ:

```
//=====
class Parents_List extends RList
{
    function FilterSql( $sql )
    {
        $sql = parent::FilterSql( $sql );
        $rvx =& get_engine();
        $grand_name = $rvx->Input->Post('grand_name');

        if( $grand_name != " )
        {
            $grand_name = urldecode( $grand_name );
            $grand_name = is_numeric( $grand_name ) ?
            ('Id='.$grand_name) : ('Name='.chr(39).$grand_name.chr(39));

            $grand_name = $rvx->Database->Retrieve('SELECT Id FROM
            Grandparents WHERE '.$grand_name);
            $ana = new RSqlAnalyzer( $sql );
            $ana->AppendFilter( 'GrandparentsId', FLD_INTEGER, $grand_name
            );

            $sql = $ana->BuildSql();
        }

        return $sql;
    }
}
//=====
```

### III. Lọc Combobox

#### Các bước thực hiện

Bước 1, 2: giống với lọc trên List.

Bước 3: Tạo **function RenderParents()**: function sẽ trả về các bản ghi có Id của bảng cùng tên với **Id** trong Combobox cha mà ta cần lọc. sự kiện được gọi khi người dùng chọn Combobox *Id của trường xảy ra sự kiện lọc*.

Ví dụ:

```
//=====
function RenderParents()
{
    return
    "
    try
    {
        var grand_id = null;

        function GrandparentsId_OnChange(combo, rec, idx)
        {
            grand_id = Ext.getCmp('GrandparentsId').getValue();
            ParentsIdStore.baseParams = {lookupkey:'Id',
            lookupname:'Name', grand_id:grand_id}
            ParentsId.setValue( " ");
            ParentsIdStore.load();
        }
        GrandparentsId.addListener( 'select',
GrandparentsId_OnChange );
    }
}
//=====
```

Bước 4: Mở file .php của model có dữ liệu được lọc, thêm **class Parents\_Model extends Rmodel** có **function RenderCombo()** để lọc câu lệnh sql. Hàm này sẽ thực hiện lọc theo dữ liệu vào ứng với trường Id, kết quả sẽ được hiển thị trong Combobox.



Ví dụ:

```
//=====
class Parents_Model extends RModel
{
    function RenderCombo()
    {
        $rvx = &get_engine();
        $searchpart = $rvx->Input->Post('query');
        $lookupkey = $rvx->Input->Post('lookupkey');
        $lookupname = $rvx->Input->Post('lookupname');
        $searchpart = strtoupper($searchpart);

        $grand_id = $rvx->Input->Post('grand_id');

        if( $grand_id )
        $grand_id = ' AND GrandparentsId='.$grand_id;

        $sql = "SELECT $lookupkey AS Id, $lookupname AS Name FROM $this-
        >TableName
        WHERE UPPER($lookupname) LIKE '$searchpart%' $grand_id
        ORDER BY $lookupname LIMIT 0, 50";

        $qry = $rvx->Database->Query( $sql );
        $data = rvx_json_encode( $qry->result_array() );
        $cb = isset($_GET['callback']) ? $_GET['callback'] : '';
        echo $cb . '({"total":' . $qry->num_rows() . ',' . "results":' . $data . '})';

    }
}
//=====
```

## IV. Tạo report thống kê

### Các bước thực hiện

Bước 1: Trong module cần báo cáo, tạo thư mục **\_reports** và copy file **reports.php** (file giống nhau và có sẵn từ trong các module đã có **\_reports**) vào module.

Bước 2: Trong thư mục **\_reports** của module cần báo cáo, tạo file **reports.xml** nếu chưa có, các file **\*.xml** là các báo cáo.

- File **reports.xml**: Định nghĩa việc hiển thị các Báo cáo có trong module.

Ví dụ:

```
//=====
<xml>
  <folders>
    <folder id="1" name="Paretn Model List " />
  </folders>
  <reports>
    <report fd="1" id="1001" name=" Paretn List " />
    ...
  </reports>
</xml>
//=====
```

- File **\*.xml**: Định nghĩa các báo cáo.

Ví dụ:

```
//=====
<xml>
<title>Parent Model</title>
<select>
  SELECT *
  FROM TempParentModel[USERID];
</select>
<fields>
  <field name="Code" type="string" />
  <field name="Name" type="string" width="150" align="C" />
  <field name="Birthday" type="date" width="150" hidden="true" />
  <field name="Sex" type="string" width="150" />
  <field name="Age" type="integer" width="150" />
  <field name="Weigth" type="money" width="100" align="C" />
  <field name="Description" type="string" width="150" />
</fields>
<params>
<param control="lookup" type="string" name="MaDanToc"
lookupmodel="quanlithuvien/dan_toc" lookuptable="DanToc" lookupkey="MaDanToc"
lookupname="MaDanToc" />
</params>
<initsql>
  DROP TABLE IF EXISTS TempParentModel[USERID];
  CREATE TABLE TempParentModel[USERID](
```

```
Code VARCHAR(32),
Name VARCHAR(64),
Sex SMALLINT,
Birthday DATE,
Age INTEGER,
Weigth NUMERIC(5,2),
Description DATE);
```

```
INSERT INTO TABLE TempParentModel[USERID]
SELECT P.Code, P.Name, CASE WHEN P.Sex = 1 THEN 'Nam' ELSE THEN 'Nữ' Sex,
P.Birthday, P.Age, P.Weight, P.Description
FROM TempParentModel P
WHERE P. Birthday BETWEEN [FROM] AND [TO]
ORDER BY P.Name;
</initsql>
</xml>
//=====
```

## V. Tạo dashboard

### Các bước thực hiện

Bước 1: Trong module cần tạo dashboard, tạo thư mục ***\_dashboard*** và file ***dashboard.php*** (file giống nhau và có sẵn từ trong các module đã có ***\_dashboard***)

Bước 2: Tạo file ***\*.xml***: Định nghĩa các báo cáo

Ví dụ:

```
//=====
<xml>
  <title>Parent by age</title>
  <select>
    SELECT Name, Age
    FROM TempParentModel[USERID]
  </select>
  <chart type="line" width="910" colspan="3" />
  <initsql>
    DROP TABLE IF EXISTS TempParentModel[USERID];
    CREATE TABLE TempParentModel[USERID](
      Name VARCHAR(100), Age INTEGER);
    INSERT INTO TempParentModel[USERID]
```

```

SELECT P.Name, P.Age
FROM ParentModel P
WHERE P.Birthday BETWEEN [FROM] AND [TO]
GROUP BY P.Birthday;
</initsql>
</xml>
//=====

```

## H. Tạo module kế thừa module đã có

Khi thao tác trên RVX, một trong những công việc quan trọng là tùy chỉnh file controller để làm cho model trở nên thân thiện hơn với người dùng. RVX có một số phương thức cha có sẵn, phục vụ một số các chức năng đặc biệt. Khi cần, nhà phát triển có thể thừa kế các phương thức này như sau:

```

//=====
function Phuong_Thuc_Cha()
{
    //code xu ly
    Parents::Phuong_Thuc_Cha();
    //code xu ly
}
//=====

//=====
class Parent_Model_Line extends RList
{
    function FetchDataSend()
    {
        $rvx =& get_engine();
        foreach( $this->DataArray as & $row )
        {
            $row['AssignUserId'] = $rvx->Database->Retrieve('SELECT Username FROM User
            WHERE Id='.$row['AssignUserId']);
            $row['AssignGroupId'] = $rvx->Database->Retrieve('SELECT Name FROM
            UserGroup WHERE Id='.$row['AssignGroupId']);
        }
        parent::FetchDataSend();
    }
}
//=====

```

## I. Tạo action

### I. Action trên Form

	Công việc
1	Gán thẻ action cho file form.xml => trong đó, thuộc tính <b>url</b> có giá trị là đường dẫn đến hàm xử lý action, có dạng “[tên module] / [tên model] / hàm xử lý”; thuộc tính <b>caption</b> là tên hiển thị của action; thuộc tính <b>popup</b> thể hiện action có chạy chế độ popup
2	Viết hàm xử lý action => Tại file Controller của model đang thêm action, ta viết hàm xử lý action. Tùy vào công việc, ta viết hàm xử lý action này theo yêu cầu.

```
//=====
<actions>
  <action caption="Crate Returning Detail"
    url="library/returning/create_child" popup="1" />
</actions>
//=====
```

#### *Gán thẻ action cho form*

```
//=====
function Create_Child()
{
  $rvx-> get_engine();
  $id = $rvx->Context->GetPaten('id');
  $rvx->Router->Redirect('libaraty/returningdetail/add/parenid/'.$id);
}
//=====
```

#### *Viết hàm xử lý action*

## II. Action trên List

**Action** trong ứng dụng có chức năng Lọc (Filter) ra các bản ghi dữ liệu theo yêu cầu nào đó.

	Công việc
1	Trong module, mở file <b>‘.php’</b> của thư mục con cần tạo Action, thêm vào một class được extends từ <b>RList</b>
2	Trong class vừa tạo, thêm 1 function là <b>FilterSql(\$sql)</b> , đối của function này chính là câu lệnh <b>SELECT</b> của file <b>List.xml</b> trong thư mục con ta tạo Action, câu lệnh <b>\$sql</b> này được lấy bằng câu lệnh <b>\$sql = parent::FilterSql( \$sql );</b>
3	Thêm lệnh <b>\$action = \$rvx-&gt;Input-&gt;Post('action');</b> dùng để lấy ra Action mà người dùng chọn.
4	Tiếp là code xử lý action tùy theo yêu cầu, cuối cùng trả về câu lệnh \$sql đã được lọc.
5	Chỉnh sửa file <b>List.xml</b> của model cần tạo action. Trong file này, thêm thẻ <b>&lt;actions&gt;</b> <b>&lt;/actions&gt;</b> chứa các <b>&lt;action&gt;</b> <b>&lt;/action&gt;</b> chính là các Action ta tạo.

```
//=====
class The_Muon_List extends RList
{
    function FilterSql( $sql )
    {
        $sql = parent::FilterSql( $sql );
        $rvx =& get_engine();
        $action = $rvx->Input->Post('action');

        if( $action == 'show_male' )
        {
            $sana = new RSqlAnalyzer( $sql );
            $sana->AppendFilter( 'GioiTinh', FLD_BOOL, 1);
            $sql = $sana->BuildSql();
        }

        if( $action == 'show_female' )
        {
            $sana = new RSqlAnalyzer( $sql );
            $sana->AppendFilter( 'GioiTinh', FLD_BOOL, 0);
        }
    }
}
```

```

        $sql = $sana->BuildSql();
    }
    return $sql;
}
}
//=====

```

### Tạo Class xử lý action

```

//=====
<actions>
    <action caption="Show Male"
    url="quanlithuvien/the_muon/index/action/show_male" />
    <action caption="Show Female"
    url="quanlithuvien/the_muon/index/action/show_female" />
</actions>
//=====

```

### Gán thẻ action cho list

## III. Upload ảnh

	Công việc
1	Tại file form.xml của model cần thêm ảnh, ta thêm control Picture với type là text. Đồng thời thêm action có thuộc tính url mang giá trị là dạng “[tên module] / [tên model] / hàm xử lý”
2	<p><b>Ta lần lượt viết các hàm:</b> <i>Upload_Picture()</i>, <i>Upload_Picture_Do()</i> để có thẻ upload ảnh, hàm <i>RenderControl(\$ctrl)</i> và hàm <i>RenderPicture()</i> để xử lý việc hiện ảnh trên form.</p> <p>Hàm <i>Upload_Picture()</i> và <i>Upload_Picture_Do()</i> thuộc lớp Controller (Có lớp cha là Rcontroller)</p> <p>Hàm <i>RenderControl(\$ctrl)</i> và hàm <i>RenderPicture()</i> thuộc lớp Form (có lớp cha là RForm)</p>

```
//=====
<control box="2" field="Picture" type="text" />
//=====

//=====
<action caption="Upload picture" url="quanlithuvien/book/upload_picture" />
//=====
```

### **Thêm control và action**

```
//=====

class Item_Form extends RForm{
    function RenderControl( $ctrl )
    {
        $html = parent::RenderControl( $ctrl );

        //filter subcategories by selected category
        if( $ctrl->Name == 'Picture' )
        {
            $html = $this->RenderPicture();
        }
        return $html;

        function RenderPicture()
        {
            $url = 'pub/item/'. $this->Model->GetField('Code').'.jpg';
            if( !file_exists( $url ) )
                return "Picture = new
                Ext.BoxComponent({src:Ext.BLANK_IMAGE_URL});";
            else
                return "Picture = new
                Ext.BoxComponent({src:Ext.BLANK_IMAGE_URL,
                autoEl:{tag:'img', cls:'tng-managed-image', src:'$url'} });";
        }
    }
}

//=====
```

### **Hàm RenderControl và RenderPicture**

```
//=====
function Upload_Picture()
```



```

    {
        $rvx = & get_engine();
        $sid = $rvx->Context->GetParam('id');

        $dlg = new RDialog();
        $dlg->Init( 'Upload picture',
            base_url().'catalog/item/upload_picture_do' );
        $dlg->AddControl( CTRL_UPLOAD, 'UploadFile', "", false );
        $dlg->AddControl( CTRL_HIDDEN, 'ItemId', $sid, false );
        $dlg->AddButton( BTN_SUBMIT );
        $dlg->Render();
    }

function Upload_Picture_Do()
{
    $rvx = & get_engine();
    $sid = $rvx->Input->Post('ItemId');
    $code = $rvx->Database->Retrieve( 'SELECT Code FROM Item
    WHERE Id= '.$sid );

    if( isset( $_FILES ) == 0 )
        return rvx_error( 'Select a file to upload' );

    $src_file = $_FILES['UploadFile']['tmp_name'];
    $dst_file = 'pub/item/'.$code.'.jpg';

    if( ! move_uploaded_file( $src_file, $dst_file ) )
        return rvx_error( 'Cannot upload file: %s', $dst_file );

    $rvx->Router->Redirect( 'catalog/item/view/id/'.$sid );
}

} //=====

```

*Hàm Upload\_Picture và Upload\_Picture\_Do*

## J. Tổng kết

Tài liệu “Hướng dẫn phát triển module trên RVX Manager” đã được hoàn thành với nội dung được trình bày rõ ràng, chi tiết. Qua tài liệu này, bạn có cái nhìn khái quát nhất về Lịch sử hình thành, phát triển, các tính năng, độ ưu việt cũng như công nghệ được sử dụng của phần mềm RVX Manager. Quan trọng hơn, bạn đọc sẽ biết cách cài đặt phần mềm trên máy tính, và trực tiếp hiểu và tạo được ra các module đơn giản thông qua các khái niệm mang tính lý thuyết và kèm theo đó là ví dụ cụ thể. Với những kiến thức nền quan trọng ở mức tổng quát trên mà bạn đọc có được sẽ là sự khởi đầu tốt để bạn bước tiếp trên con đường phát triển những module phức tạp hơn trên phần mềm RVX Manager.