

MasterView FlyBy

A fast zoom past
the MasterView template system

Deb Lewis
San Diego Ruby Users
Nov. 2, 2006

MasterView & Rails

- Welcome to Rails
 - How did I get here?!
 - Oh no... Yet Another Template System
- What does MasterView do?
 - template markup notation
 - scaffold generation
 - admin controller
 - extensibility
- How does it work?
- How do I get it?

Moving to Rails

- Relatively static content sites
 - plain old (x) html and css
 - layouts => get rid of duplication!!

Moving to Rails

- Relatively static content sites
 - plain old (x) html and css
 - layouts => get rid of duplication!!
- Simple site services
 - user accounts for site members
 - database-backed content
 - user services (forms)
 - upgrade an old .asp site

Comforts & Necessities

- Pages designed to web standards
 - xhtml and css
 - Tidy-able and validatable **at design time**
- Design-time preview
 - Reasonable fidelity in static page view
 - Dynamic content placeholders
- Collaboration - multiple site contributors
 - diverse skill levels (designers, real people)
 - allow standard html editing tools (wysiwyg)

Standard Rails Templates

- .rhtml files in app/view
- layouts, views, partials
 - factored (DRY) – good
 - html is fragmented - bad
- Page markup notation
 - `<% ... %>` notation for dynamic elements
 - ruby/rails... comfortable [for a programmer]
 - not handled well by some editors

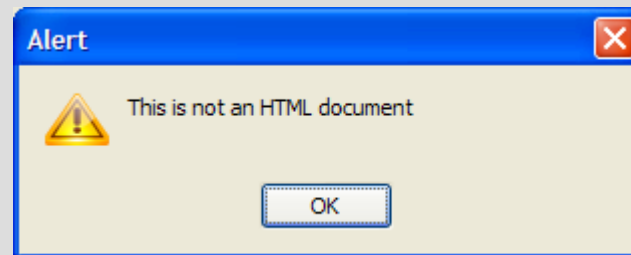
Text-based Xhtml/CSS Editor

The screenshot displays the TopStyle Pro 3.12 application window, which is a text-based Xhtml/CSS editor. The main window title is "TopStyle Pro 3.12 [W:\Organizations\xRuby\MasterView_FlyBy_20061102\app-views-rhtml\login.rhtml]". The interface includes a menu bar (File, Edit, Preview, View, HTML, Tools, Options, Help) and a toolbar with various icons for file operations, editing, and previewing. The main editing area shows the following Xhtml code:

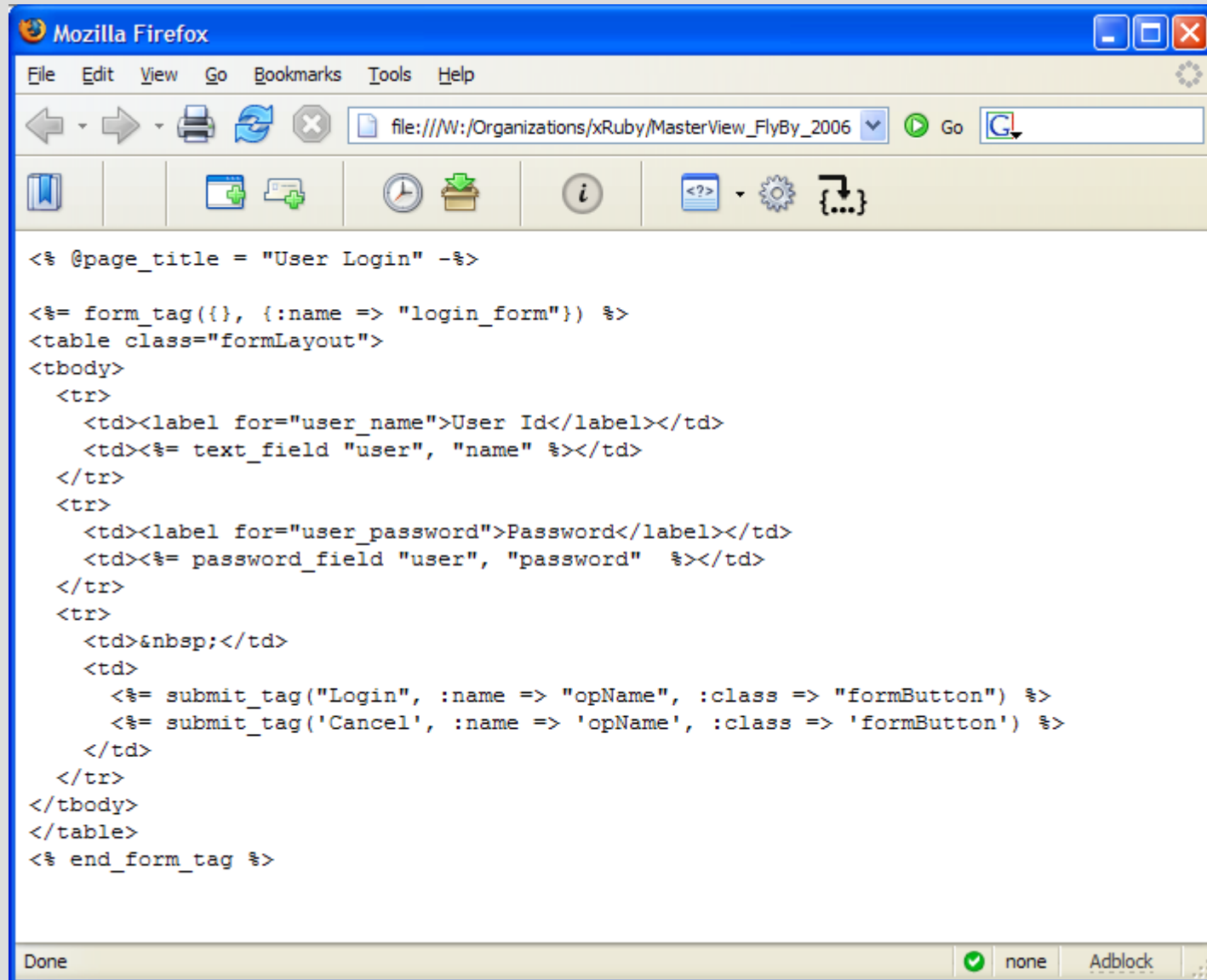
```
1 <% @page_title = "User Login" -%>
2
3 <%= form_tag({}, {:name => "login_form"}) %>
4 <table class="formLayout">
5 <tbody>
6 <tr>
7 <td><label for="user_name">User Id</label></td>
8 <td><%= text_field "user", "name" %></td>
9 </tr>
10 <tr>
11 <td><label for="user_password">Password</label></td>
12 <td><%= password_field "user", "password" %></td>
13 </tr>
14 <tr>
15 <td>&nbsp;</td>
16 <td>
17 <%= submit_tag("Login", :name => "opName", :class => "formButton") %>
18 <%= submit_tag('Cancel', :name => 'opName', :class => 'formButton') %>
19 </td>
20 </tr>
21 </tbody>
22 </table>
23 <% end_form_tag %>
```

On the right side, the Tag Inspector panel is open, showing the HTML 4.01 DTD and a list of attributes for the selected "formLayout" class. The attributes listed include: align, background, bgcolor, border, cellpadding, cellspacing, dir, frame, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, and rules. The status bar at the bottom indicates "Ln 4, Col 21".

WYSIWYG Editor (Nvu)



Design-Time Preview



```
<% @page_title = "User Login" -%>

<%= form_tag({}, {:name => "login_form"}) %>
<table class="formLayout">
<tbody>
<tr>
<td><label for="user_name">User Id</label></td>
<td><%= text_field "user", "name" %></td>
</tr>
<tr>
<td><label for="user_password">Password</label></td>
<td><%= password_field "user", "password" %></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<%= submit_tag("Login", :name => "opName", :class => "formButton") %>
<%= submit_tag('Cancel', :name => 'opName', :class => 'formButton') %>
</td>
</tr>
</tbody>
</table>
<% end_form_tag %>
```

Done none Adblock

My Rails Template Hunt

- What I wanted: Zope Page Templates
 - content written as complete (x)html documents
 - dynamic markup written in element attributes
 - macro facility for layouts, content chunks
 - Page composition markup written in attributes
- Why I like these properties
 - plays nice in variety of editors
 - design-time tools: Tidy, W3C Validator
 - reasonable fidelity previewing content chunks

My Rails Template Hunt

- What I found:
 - rhtml
 - programmer-oriented notations
(some very cool, but a different problem)
 - little or nothing with more declarative notation

My Rails Template Hunt

- What I found:
 - rhtml
 - programmer-oriented notations
(some very cool, but a different problem)
 - little or nothing with more declarative notation
- Now what?
 - port zpt's to rails [daunting]
 - give up in despair

My Rails Template Hunt

- What I found:
 - rhtml
 - programmer-oriented notations
(some very cool, but a different problem)
 - little or nothing with more declarative notation
- Now what?
 - port zpt's to rails [daunting]
 - give up in despair
- And then I found... MasterView

MasterView Template Engine

- Yes... the world needs...
Yet Another Template Engine
- Jeff Barczewski – St. Louis Ruby users
- Goals:
 - xhtml and wysiwyg editor friendly
 - previewable outside rails; designer-friendly
 - natural and appropriate in Rails
 - better scaffold generation
 - Perform as well as erb in .rhtml

MasterView Markup

- Directives written in element attributes
 - **mv:** namespace prefix
 - equivalent power to erb markup
- Built-in directives
 - Ruby expressions (emit content, view state)
 - conditionals and iteration
 - Asset helpers (stylesheets, img, links)
 - form helpers
 - Composition: layouts, content, partials
- Extensible – user-defined directives

Text-based Xhtml/CSS Editor

The screenshot displays the TopStyle Pro 3.12 interface editing a file named 'login.html'. The main editor shows the following XHTML code:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5 <head>
6 <title>Login</title>
7 <!-- design-time stylesheet references -->
8 <link href="xstyles/site.css" rel="stylesheet" type="text/css" />
9 </head>
10
11 <body>
12 <div id="PageHeader">### Page Header (banner and nav areas) ###</div>
13 <div id="PageBody">
14 <div id="PageContent">
15 <div mv:generate="{template_path}" mv:omit_tag="">
16
17   {{{ @page_title = "User Login" -}}}
18
19 <form name="login_form" mv:form="">
20 <table class="formLayout">
21 <tbody>
22 <tr>
23   <td><label for="user_name">User Id</label></td>
24   <!--<%= text_field "user", "name" %>-->
25   <td><input type="text" mv:text_field="user, name" /></td>
26 </tr>
27 <tr>
28   <td><label for="user_password">Password</label></td>
29   <!--<%= password_field "user", "password" %>-->
30   <td><input type="password" mv:password_field="user, password"/></td>
31 </tr>
32 <tr>
33   <td>&nbsp;</td>
34   <td>
35     <!--<%= submit_tag("Login", :name => "opName", :class => "formButton") %>-->
36     <input name="opName" type="submit" class="formButton" mv:submit="" value="Login" />
37     <input name="opName" type="submit" class="formButton" mv:submit="" value="Cancel" />
38   </td>
39 </tr>
40 </tbody>
41 </table>
42 </form>
43
44 </div>
```

The Tag Inspector panel on the right shows the following table of attributes for the selected 'formButton' class:

Attribute	Value
class	formButton
name	opName
type	submit
value	Login
accept	
align	
alt	
checked	<input type="checkbox"/>
dir	
disabled	<input type="checkbox"/>
id	
ismap	<input type="checkbox"/>
lang	
maxlength	
onblur	
onchange	
onclick	
ondblclick	
onfocus	
onkeydown	
onkeypress	
onkeyup	
onmousedown	
onmouseover	
onmouseout	
onmousemove	
onmouseup	
onselect	
readonly	<input type="checkbox"/>
size	
src	
style	
tabindex	
title	
usemap	

At the bottom of the window, the status bar indicates 'Ln 36, Col 52'.

WYSIWYG Editor (Nvu)

The screenshot displays the Nvu WYSIWYG editor interface. The window title is "Login [file:/.../login.html] - Nvu". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, and Help. The toolbar contains icons for New, Open, Save, Publish, Browse, Anchor, Link, Image, Table, Form, Spell, and Print. The formatting toolbar includes options for text color, background color, font size, bold, italic, underline, bulleted list, numbered list, indent, outdent, and alignment. The status bar shows "(no class)", "Variable Width", and a width of 765px. The main editing area shows a login form design with a header, a title placeholder, a form with "User Id" and "Password" fields, "Login" and "Cancel" buttons, and a footer. The status bar at the bottom shows "Normal", "HTML Tags", "Source", and "Preview" modes, with the current mode being "Normal".

Login [file:/.../login.html] - Nvu

File Edit View Insert Format Table Tools Help

New Open Save Publish Browse Anchor Link Image Table Form Spell Print

Body Text

(no class) Variable Width

765px

Page Header (banner and nav areas)

{{{ @page_title = "User Login" - }}}}

User Id

Password

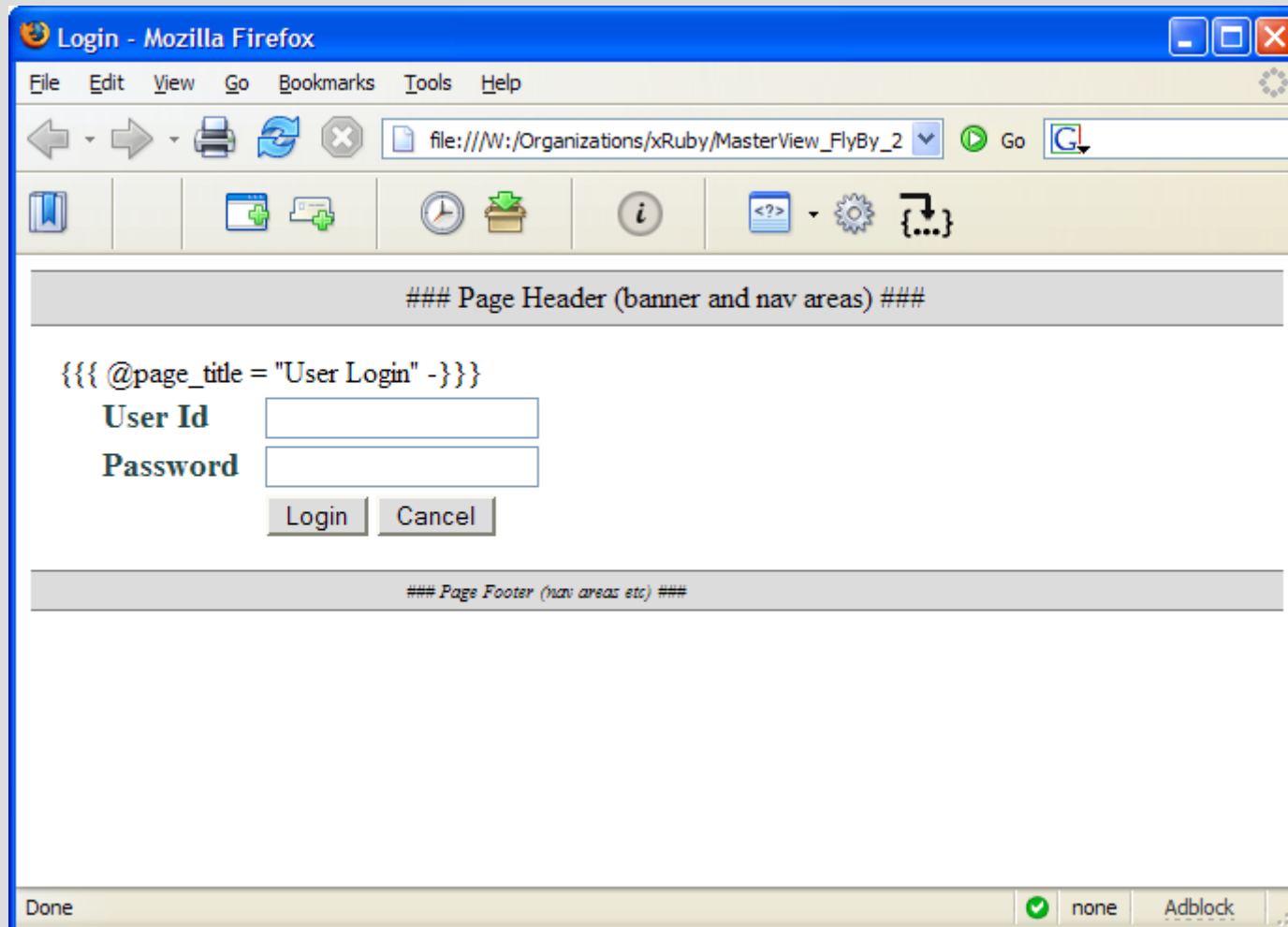
! Login Cancel

! ### Page Footer (nav areas etc) ###

Normal HTML Tags Source Preview

<body>

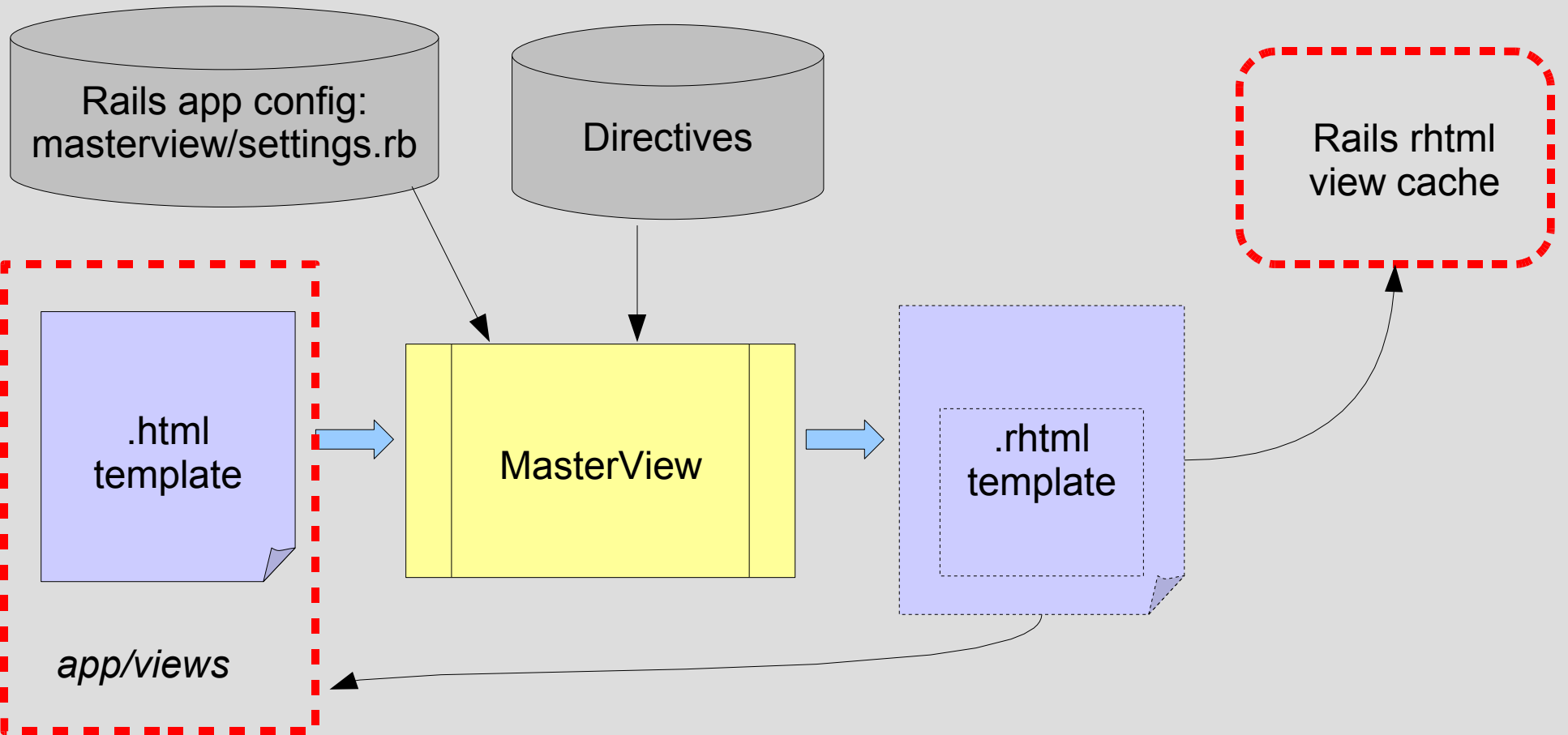
Design-Time Preview



Other Features

- Scaffold generation
 - scaffold view generator as in rails
 - goal: production-ready results
- admin control panel
 - show you templates in live site
 - live edit scratchpad
 - config info
- Extensible directive architecture
 - DSL for attribute notation, metadata
 - event model and framework for processing

How Does It Work?



How Do I Get It?

- Rails plugin – installed as gem or plugin

```
gem install masterview_gem_pack
```

```
script/plugin install  
svn://rubyforge.org/var/svn/masterview/tags/masterview
```

- <http://masterview.org>
 - documentation
 - intro videos
- Rubyforge project – downloads, lists
<http://rubyforge.org/projects/masterview>

Is It Ready?

- We think/hope so!
 - core template mechanisms solid
 - significant effort in extensibility, configurability
 - documentation
- Future:
 - more helper directives (ajax)
 - Better scaffold generation
 - admin console
- Pending: Release 0.3.0

The End

The End is just another Beginning

<http://www.masterview.org>

<http://rubyforge.org/projects/masterview>

dlewis@glaivestone.com