

HostPay:

Page files & variables documentation

Last updated 15/06/2011

Please note: We are constantly updating and improving HostPay. This may mean we add new functionality that is not listed here yet.

Table of contents

Ruby class preferences

Static pages

hosting.html	3
more_info_domains.htm	3
features.htm	3
latest_news.htm	3
about_us.htm	3
contact.htm	3
testimonials.htm	3
cvv2.html	3
service-popup.htm	3

Dynamic pages

4

Appear on all pages

header.rhtml	4
footer.rhtml	4

Non-control panel pages

index.rhtml	4
dac.rhtml	5
package-chooser.rhtml	10
domain-needed.rhtml	12
add-ons.rhtml	13
basket.rhtml	16
login-reminder.rhtml	18
register.rhtml	18
customise-contact.rhtml	19
payment-options.rhtml	19
result-complete.rhtml	22
result-abort.rhtml	23
result-cheque.rhtml	23
result-invoice.rhtml	24
dedicated-servers.rhtml	24
addon-upsell.rhtml	28
remind-password.rhtml	30
noprefs.rhtml	31
pg-lock.rhtml	31
terms.rhtml	31
template-error.rhtml	31

Control panel pages

front-page.rhtml	31
support-database.rhtml	32
header_printable.rhtml	35
footer_printable.rhtml	35
your-account.rhtml	35
invoices.rhtml	35
change-password.rhtml	39

update-contact.rhtml	40
services-domains.rhtml	41
services-hosting.rhtml	44
services-add-ons.rhtml	45
services-dedicated-servers.rhtml	45
contact-support.rhtml	45
renew.rhtml	48
renew-payment.rhtml	51
set-security-question.rhtml	52
payment-verification.rhtml	52
payment-notenrolled.rhtml	52
payment-verified.rhtml	52
payment-notify.rhtml	52
invoice_header_printable.rhtml	53
invoice_footer_printable.rhtml	53

Ruby variable availability

54

Ruby class references

String: <http://www.ruby-doc.org/core/classes/String.html>

Array: <http://www.ruby-doc.org/core/classes/Array.html>

Hash: <http://www.ruby-doc.org/core/classes/Hash.html>

ERB: <http://www.ruby-doc.org/stdlib/libdoc/erb/rdoc/index.html>
(Ruby templating)

Object: <http://www.ruby-doc.org/core/classes/Object.html>

Fixnum: <http://www.ruby-doc.org/core/classes/Fixnum.html>

Float: <http://www.ruby-doc.org/core/classes/Float.html>

Static Pages

Static pages are not processed by the ruby code so no ruby variables can be used here.
As a consequence, none of your configuration settings can be automatically passed to these pages.

Hosting.html

On this page you can list details about the hosting packages you wish to offer for sale to your customers. Before your customers can order these you should first configure them here:

<https://customer.heartinternet.co.uk/manage/manage-package-types.cgi> (Hosting Packages).

Then you should configure HostPay specific details such as the price, renewal frequency etc. here:

<https://customer.heartinternet.co.uk/manage/reseller-shop-products.cgi> (Hosting Products).

more_info_domains.htm

On this page you can list details of the domain types you wish to offer for sale to your customers.

Before your customers can order domains you should configure what types are for sale and how much they cost here: <https://customer.heartinternet.co.uk/manage/reseller-shop-domain.cgi> (Domain Products).

Features.htm

On this page you can write 'Feature' articles to highlight the benefits and uses of your products or to describe some of the features available through your hosting packages.

latest_news.htm

On this page you can keep your customers informed of any changes you make to their services e.g. special offers, price changes, changes to the features that are included in hosting packages etc. or news about your business.

about_us.htm

On this page you can give your customers some background about your company, its history, location, special achievements etc.

contact.htm

On this page you can provide your customers with contact details such as support email, address, sales email etc.

testimonials.htm

On this page you can demonstrate to potential new customers how satisfied existing customers are with the service they receive.

Cvv2.html

This page describes the 'CVV2' number on a credit card. It should not be necessary for you to edit this file.

service_popup.htm

This page describes 'Verified by Visa', which is associated with the 3D secure card verification process (used by the PayPal Pro payment gateway). It is not be necessary for you to edit this file.

Dynamic Pages

Appear on all pages

Header.rhtml

This gets prepended to every dynamic page. It contains the title, logo, telephone number, and several links. At a minimum, you are likely to want to change the title, logo, and telephone number.

For some themes this template includes the following:

```
<body
    <% if(@cgi.path_info == '/' || @cgi.path_info == '/index') %>
        id="home"
    <% end %>
>
```

This is used to modify the appearance of the 'Home' tab in the navigation bar. An example of the relevant lines in theme.css that depend on this is:

without home:

```
ul#navigation li a {height: 25px; padding-top: 10px; text-decoration:
none; color: #fff; display: block; background: #00CCFF url(../theme-
images/bg_nav.png) repeat-x top; border: 1px solid #00C5F6; border-
bottom: none;}
```

with home:

```
#home ul#navigation li.navhome a {background: #fff url(../theme-
images/bg_generic_fade.png) repeat-x top; border: 1px solid #fff; border-
bottom: none; color: #444;}
```

Footer.rhtml

This gets appended to every dynamic page. It contains links to static pages and the control panel 'home' page (front-page). You may not need to change this. Also note that you should NOT alter this line at the bottom:

```
<!-- t:20 -->
```

This identifies which theme you are currently using.

Non-Control Panel pages

Index.rhtml

This is the first page that gets displayed to your potential customers when they navigate to your HostPay installation.

Ruby Variables:

Variable	Class:Usage	Value
url_prefix	String:<%= url_prefix %>	The url of your HostPay installation eg: https://web111.secure-secure.co.uk/somedomain.co.u

Variable	Class:Usage	Value
		k/examplehostingshop/
username	String:<%= username %>	The username (email address) of the logged in user or the empty string ("")
domain_types	Array:<% domain_types.each do dt %><%= dt %><% end %>	A list of top level domain types that you have configured to be for sale eg. .co.uk, .com, .info
ds_products_for_sale?	:<% if ds_products_for_sale? %>Order Dedicated Servers<% end %>	Boolean – true if you have configured one or more dedicated server products for sale, false otherwise.

Optional code snippets:

i) A domain search widget

```
<form action='dac'>
Domain Search:<input type="text" class="searchdom" value="www."
onclick="this.value=''" name="domain"/>
<select name='tld'>
<% if domain_types.size > 1 %>
<option value=''>( all )</option>
<% end %>
<% domain_types.each do |dt| %>
<option value="<%= dt %>"><%= dt %></option>
<% end %>
</select>
<input type='submit' value='Search' />
</form>
```

Ruby code in template:

None.

dac.rhtml

This page offers your customers the ability to search for domain names they may be interested in and to find out whether they are available to register.

Ruby Variables:

Variable	Class:Usage	Value
user_menu	ERB:<%= user_menu %>	This will include (be replaced by) the template 'user-menu.rhtml' in the current location in the current template. [1]
no_upsell?	:<% if no_upsell? %>no upsell<% else %>yes upsell<% end %>	Boolean – true if no upsell has been specified, false otherwise. [2]

Variable	Class:Usage	Value
logged_in?	:<% if !logged_in? %>Log in<% end %>	Boolean – true if a user is currently logged in, false otherwise.

Notes:

[1] The user-menu template gets processed by the ruby templating code so ruby variables will be interpolated. E.g. if you add this to your menu:

```
<% if logged_in? %><li>Hi <%= username %></li><% end %>
```

The username will be inserted in the menu if the user is logged in.

[2] The no_upsell flag is used to indicate that the user is not in the normal ordering 'flow'. The normal ordering 'flow' comprises:

- Search for a domain
- Add one or more domains to your basket (a basket is created if one does not already exist)
- Optionally add a hosting package to your basket
- Optionally add one or more add-ons to your basket
- View your basket

The no_upsell flag would be set to true if the user clicked 'Order Hosting' from their basket view, for example and with the flag set the process would then return to the basket view rather than offering web hosting.

Optional code snippets:

i) Add a 'Log in' link if the user is not logged in:

```
<% if !logged_in? %><p>
<p>
<span id='logout-link' style='font-size: smaller'><a href='login'>[ Log
in ]</a></span>
</p>
<% end %>
```

Ruby code in the template:

Code:

```
<% if no_upsell? %><input type='hidden' name='no-upsell' value='1' /><% end %>
```

Description:

This enables the persistence of the no_upsell flag across pages.

Code:

```
<% if domain_types.size > 1 %>
<option value="">( all )</option>
<% end %>
<% domain_types.each do |dt| %>
<option value="<%= dt %>"><%= dt %></option>
<% end %>
```

Description:

This creates a list of your configured domain types for use in a select box.

Code:

```
<% if !logged_in? %><p>
<p>
```

```
<span id='logout-link' style='font-size: smaller'><a href='login'>[ Log in ]</a></span>
</p>
<% end %>
```

Description:

This checks whether the user is logged in and presents a Log in link if they are not.

dac-results.rhtml

This page displays the results of a domain name search. It lists any domains that, according to 'whois', are available to register and whose top level domain match one of those you have configured for sale.

If you have chosen to 'Allow Domain References' in Global Preferences and you have configured some domain transfer products for sale, this page will also present a list of domains that are available to transfer.

Ruby Variables:

Variable	Class:Usage	Value
domain_registration_availability	Array:<% domain_registration_availability.each do result %> <%= result %> <% end %>	A list of result objects which contain properties of the relevant product as well as the domain name.
domain_transfer_availability	Array:<% domain_transfer_availability.each do result %> <%= result %> <% end %>	A list of result objects which contain properties of the relevant product as well as the domain name.

Ruby code in template:

Code:

```
<% if domain_registration_availability.size == 0 and
domain_transfer_availability.size == 0 %>
<p>There were no domains available for the name you entered.</p>
<% else %>
...
<% end %>
```

Description:

domain_registration_availability and domain_transfer_availability are both arrays. If they are empty, their size will be 0. If there were no domains available to register and none to transfer following the search, inform the customer.

Code:

```
<% unless domain_registration_availability.empty? %>
[note]
If there were some domains available to register:
[/note]
...
<% domain_registration_availability.each do |result| %>
[note]
result now contains one of the domain registration results.
```

```

[/note]
<% product = result.product %>

<% next unless product %>
[note]
Ignore this result if there is no valid product.
[/note]
<% term = product.unit_period.term %>
[note]
unit_period is derived from your configuration of the relevant domain
product eg if you selected 'yearly' unit_period will be 1 year, if you
selected 'biyearly' unit_period will be 2 years.
term will contain the value 'years' or 'months'
[/note]
<% term_count = product.unit_period.count %>
[note]
term_count will contain a number.
[/note]
...
<input type='hidden' name='quoted_unit_price-<%= i %>' value='<%=
billing.price(product.unit_price) %>' />
[note]
Here 'i' is used to create a unique name for each input.
Billing.price will format the price so that there are 2 digits after the
decimal place.
These hidden inputs are used to submit the relevant data so that the
selected domains can be added to the basket.
[/note]
<input type='hidden' id='unit-price-<%= i %>' value='<%=
billing.price(product.unit_price, :use_tax_prefs) %>' />
<input type='hidden' name='base_renewal_period-<%= i %>' value='<%=
product.base_renewal_period %>' />
<%= result.name.html %>
<% if result.name_nice and result.name_nice != result.name %>
(<strong><%= result.name_nice.html %></strong>)<% end %>
...
<input type='hidden' name='name-<%= i %>' value="<%= result.name.html
%>" /></td>
        <td style='text-align: center'><% if
product.variable_period? %>
            <select id='multiply_renewal_period-<%= i %>'
name='multiply_renewal_period-<%= i %>'
onchange='javascript:update_total(<%= i %>, value)'+
            <% (product.min_years ..
product.variable_period_max).each do |multiplier| %>
                <% multiplied_term = term_count *
multiplier %>
                <% break if multiplied_term >10 %>
                <option value='<%= multiplier %>'
<% if product.default_period == multiplier %>selected='selected'<% end
%>><%= product.unit_period.to_s(multiplier) %></option>
                <% end %>
            </select>
            <% else %>
                <input type='hidden'
name='multiply_renewal_period-<%= i %>' value='1' />
                <%= product.unit_period %>

```

```

        <% end %>
      </td>
      <td style='text-align: center'><%=
billing.currency_symbol.html %><span id='price-<%= i %>'><%=
billing.price(product.unit_price, :use_tax_prefs) %>
        <% if product.variable_period? %> per <%=
product.unit_period.to_s_short.html %><% end %>
      </span></td>
      <td style='text-align: center'><input type='checkbox'
name='add-product' value='<%= product.specific_type.html %>:<%= i
%>:name,multiply_renewal_period,base_renewal_period,quoted_unit_price'
<% if domain_availability.size==1 %> checked='checked' <% end %>/>

<script type='text/javascript'>
if(document.getElementById('multiply_renewal_period-<%= i %>'))
  update_total(<%= i %>,
document.getElementById('multiply_renewal_period-<%= i %>').value);
</script>
    </td>

</tr>
<% i = i + 1 %>
<% end %>

```

Description:

If there were one or more domains available to register matching the search criteria, the above code will display them as rows in a table. See notes above, enclosed by [note][note].

Code:

```

<% unless domain_transfer_availability.empty? %>
<h3>Domains Available to Transfer</h3>
<p>The
  following domains are already registered, if you own any of these
  domains why not transfer it to us? Simply select the domain name
  below.</p>
<table border="1" class='dac-results-table'>
<thead>
<tr>
<th>Domain</th>
<th>Keep Nameservers?</th>
<th>Price*</th>
<th>Add to Basket</th>
</tr>
</thead>
<tbody>
<% domain_transfer_availability.each do |result| %>
<% product = result.transfer_product %>
<% next unless product %>
<% if result.transferable_with_tag_change %>
<tr><td><%= result.name.html %></td><td colspan='3'>Please change
IPSTAG</td></tr>
<% end %>
<% next if result.transferable_with_tag_change %>
<% if result.transferable_with_unlock %>
<tr><td><%= result.name.html %></td><td colspan='3'>Please unlock with
your existing registrar</td></tr>
<% end %>
<% next if result.transferable_with_unlock %>

```

```

<% term = product.unit_period.term %>
<% term_count = product.unit_period.count %>
<tr>
<td>
<input type='hidden' name='quoted_unit_price-<%= i %>' value='<%=
billing.price(product.unit_price) %>' />
<input type='hidden' id='unit-price-<%= i %>' value='<%=
billing.price(product.unit_price, :use_tax_prefs) %>' />
<input type='hidden' name='base_renewal_period-<%= i %>' value='<%=
product.base_renewal_period %>' />
<%= result.name.html %>
<% if result.name_nice and result.name_nice != result.name %>
(<strong><%= result.name_nice.html %></strong>)<% end %>
<input type='hidden' name='name-<%= i %>' value="<%= result.name.html
%>" />
<input type='hidden' name='keep_nameservers-<%= i %>'
id='keep_nameservers-<%= i %>' value="1" disabled='disabled' /></td>
<td style='text-align: center'>
<select name='keep_nameservers-<%= i %>'>
<option value="1">Yes</option>
<option value="" selected='selected'>No</option>
</select>
</td>
<td style='text-align: center'>
<% if result.is_free.nil? %>
<td style='text-align: center'>N/A</td>
<td style='text-align: center'> </td>
<% else %>
<td style='text-align: center'><%=
billing.currency_symbol.html %><%= billing.price(product.unit_price,
:use_tax_prefs) %></td>
<td style='text-align: center'><input type='checkbox' name='add-
product' onclick='javascript:return(confirm("You are adding a domain to
your basket that is already registered. You must be the owner of this
domain to be able to transfer the domain to us"))' value='<%=
product.specific_type.html %>:<%= i
%>:name,quoted_unit_price,base_renewal_period,keep_nameservers' /> </td>
<% end %>
</td>
</tr>
<% i = i + 1 %>
<% end %>

```

Description:

If there were one or more domains available to transfer matching the search criteria, the above code will display them as rows in a table.

package-chooser.rhtml

This page presents a list of web hosting packages that you have configured for sale and a list of domains belonging to the user (if ordered through hostpay) or currently in their basket so that they can select which domain the hosting package is to be associated with.

Ruby Variables:

Variable	Class:Usage	Value
domains	Array:<% domains.each do domain %> <%= domain.html %> <% end %>	Array of domains that can be linked to a web hosting package. This includes domains the customer purchased earlier

Variable	Class:Usage	Value
		and domains currently in their basket.
hosting_products	Array:<% hosting_products.each do product %> <%= product.name %> <% end %>	Array of web hosting products. Each product object has properties including name, description, base_renewal_period, and unit-price

Ruby code in template:

Code:

```
<p>This will be the name of your web site.</p>
<p>http://www.<select name='domain-'>
<% domains.each do |domain| %>
<option value="<%= domain.html %>"><%= domain.html %></option>
<% end %>
</select></p>
```

Description:

Domains is an Array of domains that could be linked to the hosting package.

The above code creates a dropdown list of domains for the customer to choose from. The selected domain will become linked to any selected hosting package on order completion.

Code:

```
<% i = 0 %>
<% hosting_products.each do | product | %>
<h4><%= product.name.html %></h4>
<p><%= product.description %></p>
<input type='hidden' name='base_renewal_period-<%= i %>' value="<%=
product.base_renewal_period %>" />
<input type='hidden' name='quoted_unit_price-<%= i %>' value="<%=
billing.price(product.unit_price) %>" />
<p><input type='radio' name='add-product' value='<%=
product.specific_type.html %>:<%= i
%>:base_renewal_period,quoted_unit_price,domain' /><%=
billing.currency_symbol.html %><%= billing.price(product.unit_p
rice, :use_tax_prefs) %>
<% if product.base_renewal_period != 'never' %>
every <%= product.unit_period.to_s_short.html %>
<% end %>
<% if product.setup_fee != 0 %>
plus a one off fee of <%= billing.currency_symbol.html %><%=
sprintf("%.2f", if billing.include_tax? then
billing.price(product.setup_fee) else product.setup_fee end) %>
<% end %>
<% if product.has_free_products? %>
<br />
includes free domains
<% product.free_products_list.each do | free_product | %>
<%= free_product["free_name"] %>
<% end %>
<% end %>
```

```
*
</p>
<% i = i + 1 %>
<% end %>
```

Description:

The above code creates a list of web hosting packages that you have configured for sale, formatted as a heading, a description paragraph, a radio to select the item, the renewal period, and optional extra text relating to setup fees and free domain types.

Domain-needed.rhtml

This page enables the customer to search for a domain name to register or (if you have enabled domain references in your configuration) enter a domain name for which a domain reference will be created.

The customer will be redirected to this page if they attempt to order web hosting but do not have any domains that could be linked to.

Ruby Variables:

Variable	Class:Usage	Value
error_message	String:<%= error_message %>	Nothing or I'm sorry, this domain name type is not supported
domain_references_allowed?	:<% if domain_references_allowed? %> You can order references <% end %>	Boolean – true if you have enabled domain references in your configuration, false otherwise.

Ruby Code in Template:

Code:

```
<% if error_message %><p class='error'><%= error_message %></p><% end %>
```

Description:

If the last action resulted in an error, inform the customer.

Possible messages: I'm sorry, this domain name type is not supported

Cause: The domain does not have one of the allowed suffixes (tlds).

Code:

```
<% if domain_references_allowed? %>
```

```
<p>or</p>
<p>Enter a domain name whose DNS records you will be maintaining with
another provider.</p>
<form accept-charset='utf-8' method='get' onsubmit="return
check_domain_input('fake-domain')" class='normal-input-width'>
<p><input type='text' size='30' name='fake-domain' /></p>
<p><input type='submit' value='Continue' /></p>
</form>
<% end %>
```

Description:

If you have enabled domain references in your configuration, the above code will add a form

allowing your customer to add a domain reference to their basket.

Add-ons.rhtml

This page enables the customer to add one or more of the add-on products that you have configured for sale to their basket.

Ruby Variables:

Variable	Class:Usage	Value
addon_products	Array:<% addon_products.each do product %> <%= product.name.html %> <% end %>	Array of web hosting products. Each product object has properties including name, description, base_renewal_period, and unit-price
basket	(Basket)Object:	A representation of the basket. Properties: items – a collection (array) of Basket::Item objects discount_total – the total discount for the basket. discount_total_withvat – the total discount for the basket including tax (vat). Basket::Item: Properties: product – a Product object name – a String, usually the associated domain name. setup_fee – a Float representing the set up fee for the product. extra_data – a Hash renewal_period – a Period object representing the renewal period. quoted_unit_price – a Float quoted_unit_price_incvat – a Float quoted_price – a Float quoted_price_incvat – a Float id – a Fixnum for internal use only. Product: Properties: unit_period – a Period object representing the base renewal

Variable	Class:Usage	Value
		period of the product. base_renewal_period – a String ('monthly', 'quarterly', 'yearly', 'biyearly', or 'never'). variable_period_max – a Fixnum. Period: Properties: term – a String (either 'month' or 'year') count – a Fixnum
subdomains	Array:<% subdomains.each do subdomain %> <%= subdomain.html %> <% end %>	Array of strings. Values: www secure webmail payment ssl order

Ruby Code in Template:

Code:

```

<% i = 0 %>
<% addon_products.each do | product | %>
<h4><%= product.name.html %></h4>
<p><%= product.description %></p>
<input type='hidden' name='base_renewal_period-<%= i %>' value="<%=
product.base_renewal_period %>" />
<input type='hidden' name='quoted_unit_price-<%= i %>' value="<%=
billing.price(product.unit_price) %>" />
<% if product.linkable? %>
<p>Link to:
<%= product.link_select(i, linkable_domains, package_linked_domains)
%></p>
<% end %>
<% if product.needs_ssl_info? %>
<div class="ssl_form">
  <label for="subdomain-<%= i %>">Subdomain: </label><select
name="subdomain-<%= i %>" id="subdomain-<%= i %>"
onchange="alter_ssl_email_message(<%= i %>);">
<% subdomains.each do |subdomain| %>
  <option value="<%= subdomain.html %>"><%= subdomain.html %></option>
<% end %>
</select>
  <input type="checkbox" name="contact_details-<%= i %>"
id="contact_details-<%= i %>" value="yes" /> <label for="contact_details-
<%= i %>">I confirm that my contact details are accurate.</label>
  <p id="ssl_email_warning-<%= i %>" class="ssl_email_warning">SSL
Certificates require email confirmation at the admin mailbox for the
subdomain, or at the domain in the case of the www subdomain.</p><script
type="text/javascript">alter_ssl_email_message(<%= i %>);</script>
</div>

```

```

<% end %>
<% if product.link_error==:no_domains %>
<p class='error'>You have no domain names available to attach this to.
Please <a href='dac'>order one</a> if you wish to purchase this item.</p>
<% elsif product.link_error==:no_packages %>
<p class='error'>You have no packages available to attach this to. Please
<a href='package-chooser'>order one</a> if you wish to purchase this
item.</p>
<% elsif product.link_error %>
<p>Purchasing this add-on is not currently possible. Please try again
later.</p>
<% end %>
<% if product.needs_email? %>
<% if not domains.empty? %>
<p>Email Address:<input type='text' name='exchange_prefix-<%= i
%>' />@<select name='exchange_domain-<%= i %>'>
<% domains.each do |domain| %>
<option value="<%= domain.html %>"><%= domain.html %></option>
<% end %>
</select></p>
<% else %>
<p class='error'>You have no domain names to put a mailbox on. Please <a
href='dac'>order one</a> if you wish to purchase this item.</p>
<% end %>
<% end %>
<p><input <% if product.link_error %>disabled='disabled' <% end
%>type='checkbox' name='add-product' value='<%=
product.specific_type.html %>:<%= i
%>:base_renewal_period,quoted_unit_price,link_to,exchange_prefix,exchange
_domain,subdomain,contact_details' /><%= billing.currency_symbol.html
%><%= billing.price(product.unit_price, :use_tax_prefs) %>
<% if product.base_renewal_period != 'never' %>
every <%= product.unit_period.to_s_short.html %>
<% end %>
<% if product.setup_fee != 0 %>
plus a one off fee of <%= billing.currency_symbol.html %><%=
sprintf("%.2f", if billing.include_tax? then
billing.price(product.setup_fee) else product.setup_fee end) %>
<% end %>
*
</p>
<% i = i + 1 %>
<% end %>

```

Description:

The above code creates a list of add-ons that you have configured for sale, formatted as a heading, a description paragraph, an optional dropdown offering domains to link to, optional elements to select the subdomain for ssl and confirm contact details, an optional section to select the email address for exchange mailbox, optional set up fee, and a checkbox to select the item and the renewal period.

Code:

```

<% unless basket.nil? or basket.empty? or no_upsell? %>
<p><a href='basket?add-no-product=add-on'>No thank you, please continue
with my order ....</a></p>
<% end %>

```

Description:

If there is a non-empty basket object and the no_upsell flag is not set, include a link to continue the order without adding an add-on product to the basket.

Basket.rhtml

This page displays the current basket contents and offers the user the opportunity to delete items from the basket, order more items, or continue with the order process.

Ruby Variables:

Variable	Class:Usage	Value
discount_codes_size	Fixnum:There are <%= discount_codes_size %> discount codes.	The number of discount codes.
google_button_image_url	String:<%= google_button_image_url %>	The url for the Google Checkout button image.
has_basket_items?	:<% if has_basket_items? %>Yes<% else %>No<% end %>	Boolean – true if the customer has some items in their basket, false otherwise.

Ruby Code in Template:**Code:**

```
<% if basket.current_items.empty? %>

<p>Your basket is currently empty.</p>
<p>Please choose from one of the following areas to view our products:-
</p>
<ul>
<li><a href='dac<% if no_upsell? %>?no-upsell=1<% end %>'>Domain
Names</a></li>
<li><a href='package-chooser<% if no_upsell? %>?no-upsell=1<% end %>'>Web
Hosting</a></li>
<% if ds_products_for_sale? %>
<li><a href='dedicated-servers<% if no_upsell? %>?no-upsell=1<% end
%>'>Dedicated Servers</a></li>
<% end %>
<li><a href='add-ons<% if no_upsell? %>?no-upsell=1<% end %>'>Add-
ons</a></li>
</ol>
<% else %>
```

Description:

If your basket is currently empty (contains no items) offer a list of links to order items.

Code:

```
<table class='basket'>
<thead>
<tr><th>Product</th><th>Period</th><th>Price *</th><th>Delete?</th></tr>
</thead>
<tfoot>
<tr><th colspan='2'>Total
<% if not billing.tax_applies? %>
<% elsif billing.include_tax? %>
    including <%= billing.tax_name %>
<% else %>
```

```

        excluding <%= billing.tax_name %>
<% end %>
</th>
<td style='text-align: center'><%= billing.currency_symbol.html %><%=
sprintf("%.2f", if billing.include_tax? then
billing.price(basket.to_f_withvat, :invert_tax_prefs) else
basket.to_f.to_money end) %></td></tr>
</tfoot>
<tbody>
<% basket.each do |item| %>
<% product = item.product %>
<tr>
<td class='item-name'>
<% if product.type.include?('package') %>Hosting: <% end %><%=
product.name.html %><% if item.name != "" %>: <%= item.name.html %><% end
%><% if item.setup_fee > 0 %><br />Includes a one off setup fee of <%=
billing.currency_symbol.html %><%= sprintf("%.2f", if
billing.include_tax? then item.setup_fee_incvat else item.setup_fee end)
%><% end %>
<% if product.has_free_products? %>
<br />includes free domains
<% product.free_products_list.each do | free_product | %>
<%= free_product["free_name"] %>
<% end %>
<% end %>
<% if (item.extra_data['link_to'] and item.name == "") %>:<%=
item.extra_data['link_to'] %><% end %>
</td>
<td><%= item.renewal_period.to_s || "N/A" %></td>
<td style='text-align: center'><%= billing.currency_symbol.html %><%= if
billing.include_tax? then item.quoted_price_incvat else item.quoted_price
end %></td>
<td><a href='basket?delete-product=<%= item.id %>;stay=1'></a></td>
</tr>
<% end %>
</tbody>
</table>

```

Description:

This is the table that presents the contents of the basket. A row is added for each item in the basket and the grand total is displayed in the footer.

Code:

```

<form id='continue_form' action='customise-contact'>
<% if discount_codes_size > 0 %>
<p>
If you have a discount code please enter it here:<input type='text'
name='discount_code' />
</p>
<% end %>
<p>
<input type='submit' value='Continue to Checkout' />
</p>
<!-- Google Checkout button -->
<% if google_button_image_url %>
<p>
<script type="text/javascript"
src="https://checkout.google.com/seller/accept/j-en_GB.js"></script>

```

```

<script type="text/javascript">showMark(2);</script>
</p>
<noscript>
<p></p>
</noscript>
<% end %>
</form>

```

Description:

This is the form that takes the user to the next page. If you have configured any discount codes the user will be able to enter one here. If you have configured the Google Checkout payment gateway an element will be added that the customer can click on to find out about Google Checkout.

login-reminder.rhtml

On this page your customer can log in if they already have an account or they can start the registration process.

Ruby Variables:

Variable	Class:Usage	Value
login_attempted	:<% if login_attempted %> Attempted <% end %>	Boolean: true if the user has attempted but failed to login, false otherwise.
target_username	String:<%= target_username.html %>	The email address that may have been entered.
discount_code	String:<%= discount_code %>	Any discount code that was entered on the previous page.

Ruby Code in Template:

Code:

```

<% if login_attempted %>
<p class='error'>The email address or password you entered couldn't be
verified. Please check that both were entered correctly.</p>
<% end %>

```

Description:

If the user attempted to login but failed, inform them.

Register.rhtml

On this page your customer can register their details to create a new account. Some validation is performed so that the chances of successful domain registration are higher.

Note that this page depends on 'default.css' and 'default.js'.

Ruby Variables:

Variable	Class:Usage	Value
error	String:<%= error %>	A value indicates that it was not possible to create the new account.
field	(method call):<%= field('email') %>	Inserts the relevant html to implement the specified field.

Ruby Code in Template:

Code:

```
<% if error %><p class='error'>An internal error occurred! Please try again later</p><% end %>
```

Description:

If it was not possible to create the new account, inform the user.

Code:

```
<tr><th class='normal-input-width'>Salutation</th><td> <%= field('salutation') %> </td></tr>
<tr><th>First Name(s)</th><td><%= field('name:first') %></td></tr>
```

Description:

Add the input fields so that the user can enter their registration data.

customise-contact.rhtml

If the basket contains a domain, it is possible that the domain is to be registered to an address other than that used during registration. This page offers the user the chance to enter the specific details that this domain should be registered to.

Ruby Code in Template:

Code:

```
<tr><th class='normal-input-width'>Salutation</th><td> <%= field('salutation') %> </td></tr>
<tr><th>First Name(s)</th><td><%= field('name:first') %></td></tr>
```

Description:

Add the input fields so that the user can enter their contact data.

Payment-options.rhtml

On this page the customer can choose how to pay for the items in their basket. They are presented with the contents of their basket and offered a choice of payment methods.

Ruby Variables:

Variable	Class:Usage	Value
paypalpro_page_style	String:<%= paypalpro_page_style %>	If you have created a unique named page style in PayPal for PayPal Pro payments, and applied it to your payment gateway settings, it will be passed in the form to PayPal via this variable.
gateways	Array:<% gateways.each do g %> <%= g.to_html %> <% end %>	An array of PaymentGateway objects. Properties: to_html – a string containing an html form with all of the inputs required for submission to the payment gateway provider, pre-populated with the relevant data.


```

        else
            item.quoted_unit_price
        end
    end) %>
</td>
<td style='text-align: right'><%= billing.currency_symbol.html %><%=
sprintf("%.2f",
    if billing.include_tax? then
        if (item.discount_percent > 0) then
            item.discounted_quoted_price_incvat
        else
            item.quoted_price_incvat
        end
    else
        if (item.discount_percent > 0) then
            item.discounted_quoted_price
        else
            item.quoted_price
        end
    end) %>
</td>
</tr>
<% end %>

```

Description:

This displays a representation of the contents of the basket so that the customer can see what they are about to pay for.

Code:

```

<% if basket.discount_total > 0 %>
<tr><td colspan='4'>Discount</td><td style='text-align: right'><%=
billing.currency_symbol.html %><%= sprintf("%.2f", if
billing.include_tax? then basket.discount_total_withvat else
basket.discount_total end) %></td></tr>
<% end %>

```

Description:

This displays the amount of any discount.

Code:

```

<% if basket.to_f > 0 then %>
    <## Total > 0 %>
    <% gateways.each do |g| %>
        <%= g.to_html %>
    <% end %>
    <% if gateways.size > 1 %>
        <p>Please select the payment provider you wish to
use.</p>
    <% end %>

    <% if (google_button_image_url and gateways.size > 1) or (not
google_button_image_url) then %>
        <% if gateways.size == 1 and not google_button_image_url
%>
            <input type='hidden' name='chosen_gateway'
value='<%= gateways[0].name.html %>' id='cg' />
            <script
type='text/javascript'>select_form(document.getElementById("cg"));</scrip
t>

```

```

        <% else %>
            <select id="chosen-gateway" name='chosen_gateway'
onchange='select_form(this);'>
                <% if dropdown_gateways > 1 %><option
value=''>Please select one</option><% end %>
                <% gateways.each do |g| %>
                    <% if not g.to_option.include?
"GoogleCheckout" and not g.to_option.include? "PayPalProDirect" then %>
                        <%= g.to_option %>
                    <% end %>
                <% end %>
            </select>
            <%# If there is only 1 gateway, do select_form
for it %>
                <% if dropdown_gateways == 1 %>
                    <script type='text/javascript'>

select_form(document.getElementById("chosen-gateway"));
                    </script>
                <% end %>

        <% end %>
        <input type='hidden' id='form_to_submit' value='' />
        <input type='button' id='sub_button' value='Complete
Order' onclick='submit_form();' />
        <% end %>
        <% if google_button_image_url %>
            <% if gateways.size > 1 %>
                <p>Or use</p>
            <% end %>
            <input type='image' src='<%= google_button_image_url.html
%>' alt='<%= google_button_image_alt %>' onclick='javascript:
select_form_by_name("GoogleCheckout"); return submit_form();' />
        <% end %>

<% if paypalpro_page_style %>
<% if gateways.size > 1 %>
<p>Or use</p>
<% end %>

```

Description:

This displays the list of payment gateways that you have configured, enabling the customer to select one.

result-complete.rhtml

This page is displayed to your customers following successful payment.

Note that this snippet:

```
<WPDISPLAY ITEM=banner>
```

is only required for WorldPay payments.

Ruby Variables:

Variable	Class:Usage	Value
invoice_number	Fixnum:<%= invoice_number %>	Will contain the invoice number related to the order just processed, where available.

Ruby Code in Template:**Code:**

```
<% if invoice_number %>
<p>
Please click here to view your invoice <a target='_blank' href='<%= url
%>invoices?num=<%= invoice_number %>'>View Invoice</a>.
</p>
<p>
Please click here to print / download your invoice <a target='_blank'
href='<%= url %>invoices?num=<%= invoice_number
%>;printable=1'>Print/Download Invoice</a>.
</p>
<% end %>
```

Description:

If the invoice number is available, include links to view the invoice.

Result-abort.rhtml

This page is displayed to your customers if they abort the payment process or the payment failed.

Result-cheque.rhtml

This page is displayed to your customers if they selected the 'Cheque' payment gateway. It provides details of how to make the payment e.g. where to send the cheque to and who to make it out to.

Ruby Variables:

Variable	Class:Usage	Value
address	String:<%= address.gsub(/[\r\n]+/, ' ') %>	The address to which the cheque should be posted. This is the address you applied to your payment gateway configuration for 'Cheque' gateway.
payable	String:<%= payable.gsub(/[\r\n]+/, ' ') %>	To whom the cheque should be payable.
amount	String:<%= amount %>	The amount that the cheque should be for.

Ruby Code in Template:**Code:**

```
<% if address %>
<%= address.gsub(/[\r\n]+/, '<br />') %>
<% else %>
<strong>Postal Address</strong>
<% end %>
```

Description:

If an address has been configured it will be inserted here.

Code:

```
<% if payable %>
<%= payable.gsub(/[\r\n]+/, '<br />') %>
<% else %>
```

Name

<% end %>

Description:

If a 'payable to' has been configured it will be inserted here.

Code:

<p>

<%= billing.currency_symbol.html %> <%= amount %>

</p>

Description:

This will display the amount that the cheque was for, preceded by a currency symbol e.g. £

Result-invoice.rhtml

This page is displayed to your customers if they selected the 'Invoice' payment gateway. It provides details of how to make the payment e.g. instructions and an email address to send confirmation once the invoice has been paid.

Ruby Variables:

Variable	Class:Usage	Value
invoice_instructions	String:<%= invoice_instructions.gsub(/[\r\n]+/, ' ') %>	Text instructions detailing how the invoice should be paid.
invoice_email	String:<%= invoice_email %>	An email address to which confirmation of payment should be sent.

Ruby Code in Template:

Code:

<% if invoice_instructions %>

<h3>Instructions</h3>

<p>

<%= invoice_instructions.gsub(/[\r\n]+/, '
') %>

</p>

<% end %>

Description:

If instructions have been set in the configuration, display them.

Code:

<% if invoice_email %>

<p>

Once you have made payment

please email confirmation to <a href="mailto:<%= invoice_email %>"><%= invoice_email %>.

</p>

<% end %>

Description:

If an email address has been configured, display it.

dedicated-servers.rhtml

If you have configured dedicated servers for sale, your customers can select what type of server they would like to order on this page.

Ruby Variables:

Variable	Class:Usage	Value
dual_core_for_sale	:<% if dual_core_for_sale %> Dual Core <% end %>	Boolean: true if you have configured one or more dual core dedicated servers for sale.
quad_core_for_sale	:<% if quad_core_for_sale %> Quad Core <% end %>	Boolean: true if you have configured one or more quad core dedicated servers for sale.
ds_spec_details	Hash:<%= ds_spec_details['2c']['title'] %>	Keys are '2c' and '4c'. In this usage, '2c' represents dual core ('4c' would represent quad core) and title will result in 'Dual Core Dedicated Server' at time of writing.
os_to_name	Hash:<%= os_to_name['linux'] %>	Linux -> Linux (CentOS), windows -> Windows Server 2008 Web Edition, windows_se -> Windows Server 2008 Standard Edition
os_list	Hash:<% os_list['2c'].each_key do os_key %> <%= os_key %> <% end %>	Keys are '2c' and '4c'. The values associated with they keys are arrays of strings. This is used in conjunction with the os_to_name hash to generate descriptions of the operating system e.g. 'Linux (CentOS)' etc.
default_cpanel_price	Hash:cPanel Type is <%= default_cpanel_price['type_per iod'].split('-')[0] %> cPanel Renewal frequency is <%= default_cpanel_price['type_per iod'].split('-')[1] %> cPanel Price is <%= sprintf("%.2f", default_cpanel_price['price']) %>	The 'type' is for internal use only. The renewal frequency (period) will be either 'monthly' or 'yearly'. The price is the price excluding tax.
default_backup_price	Array:<% default_backup_price.each do backup_data %> Backup size is <%= backup_data['backup_size'] %> Backup price is <%= sprintf("%.2f",	The 'type' is for internal use only. The renewal frequency (period) will be either 'monthly' or 'yearly'. The price is the price excluding tax.

Variable	Class:Usage	Value
	<pre> backup_data['price']) %> Backup Type is <%= backup_data['type_period'].spli t('-')[0] %> Backup Renewal frequency is <%= backup_data['type_period'].spli t('-')[1] %> </pre>	The size is the backup size in GB.
free_products	<pre> Hash:<% free_products.each_key do fp %> <%= free_products[fp] %>
 <% end %> </pre>	The key is a 'free id' e.g. 'domain.gtld'. The value is a 'name' e.g. 'Global Domain (.com etc)'.

Ruby Code in Template:

Code:

```

<% if dual_core_for_sale %>
<td>
<h3><%= ds_spec_details['2c']['title'] %></h3>
<p>
Processor (CPU): <%= ds_spec_details['2c']['cpu'] %><br />
Hard Drive: <%= ds_spec_details['2c']['hdd'] %><br />
Memory (RAM): <%= ds_spec_details['2c']['ram'] %><br />
Bandwidth: <%= ds_spec_details['2c']['bandwidth'] %>
</p>

```

Description:

If you have configured one or more Dual core dedicated servers for sale, this code will display the 'name' eg. Dual Core Dedicated Servers' followed by the product details: CPU, hard drive, memory, and bandwidth.

Code:

```

<% if os_list['2c'].size != 1 %>
    <p>Select Operating System:</p>
    <select name="os" id="2c-os" onchange="javascript:
update_cpanel_and_radios('2c');">
        <% os_list['2c'].each_key do |os_key| %>
            <option value="<%= os_key %>"><%=
os_to_name[os_key] %></option>
        <% end %>
    </select>
<% else %>
    <% os_list['2c'].each_key do |os_key| %>
        <p>Operating System: <%= os_to_name[os_key] %></p>
        <input type="hidden" name="os" id="2c-os" value="<%=
os_key %>" />
    <% end %>
<% end %>

```

Description:

If you have configured more than 1 dual core dedicated server type for sale, the customer will be

presented with a drop down to choose one, otherwise they will just be informed what it is.

Code:

```
<% unless default_cpanel_price.empty? %>
  <p>Add cPanel?</p>
  <select name="cpanel" id="2c-cpanel" onchange="javascript:
add_or_remove_cpanel('2c');">
    <option value="">No</option>
    <option value="<%= default_cpanel_price['type_period']
%>" id="2c-cpanel-yes">Yes - +<%= billing.currency_symbol.html %><%=
sprintf("%.2f",billing.price(default_cpanel_price['price'],
:use_tax_prefs)) %>* per <% if
default_cpanel_price['type_period'].match(/monthly/) %>month<% else
%>year<% end %></option>
  </select>
<% end %>
```

Description:

If you have configured any cPanel products for sale, this will offer the customer the chance to add one to their order.

Code:

```
<% unless default_backup_price.empty? %>
  <p>Add Server Backups?</p>
  <select name="backup" id="2c-backup" onchange="javascript:
add_or_remove_backup('2c');">
    <option value="">No</option>
    <% default_backup_price.each do |backup_data| %>
      <option value="<%= backup_data['type_period'] %>"
id="2c-backup-size"><%= backup_data['backup_size'] %>GB +<%=
billing.currency_symbol.html %><%=
sprintf("%.2f",billing.price(backup_data['price'], :use_tax_prefs)) %>*
per <% if backup_data['period'].match(/monthly/) %>month<% else %>year<%
end %></option>
    <% end %>
  </select>
<% end %>
```

Description:

If you have configured any backup products for sale, this will offer the customer the chance to add one to their order.

Code :

```
<% if free_products.size > 0 %>
<p>
If you order a Dedicated Server you will be entitled to one free
registration<br />for the minimum period of a domain of one of the
following types:<br />
<% free_products.each_key do |fp| %>
<%= free_products[fp] %><br />
<% end %>
</p>
<% end %>
```

Description:

If there are any free products associated with your dedicated server products, this will inform the customer what they are.

Addon-upsell.rhtml

If you have enabled upsell of add-ons in your global preferences and have configured some of your add-on products to be upsold with a specific product, once the correct type of product is in your customer's basket, this page will offer them the chance to add some more products to their basket.

Ruby Variables:

Variable	Class:Usage	Value
upsellable_item_list	Array:<% upsellable_item_list.each do item %> <% product = item.product %> <%= product.name %> <% end %>	This is an array of basket items for which something can be upsold with their associated product.
addon_name_with_to_upsell_text	Hash:<% addon_name_with_to_upsell_text.each_key do aname_with %> Name _ with is <%= aname_with %> Upsell text is <%= addon_name_with_to_upsell_text[aname_with] %> <% end %>	The key is composed of the addon 'name' joined to either 'domain' or 'package' by '_'. The value is the 'upsell text' – some text to try to convince the customer that they should buy the extra product.
upsold_name_to_domain	Hash:<% upsold_name_to_domain.each_key do name %> Already upsold addon name is <%= name %> <% end %>	The key is an addon name. The value is a domain.

Ruby Code in Template:

Code:

```
<% upsellable_item_list.each do |item| %>  
  <% product = item.product %>  
<!-- start of product description -->  
  <% if product.type.include?('package') %>Hosting: <% end %><%=  
product.name.html %><% if item.name != "" %>: <%= item.name.html %><% end  
><% if item.setup_fee > 0 %><br />Includes a one off setup fee of <%=  
billing.currency_symbol.html %><%= sprintf("%.2f", if  
billing.include_tax? then item.setup_fee_incvat else item.setup_fee end)  
><% end %>  
  <% if product.has_free_products? %>  
<br />includes free domains  
    <% product.free_products_list.each do | free_product | %>  
<%= free_product["free_name"] %>  
    <% end %>  
  <% end %>  
  <% if item.extra_data['link_to'] %> linked to <%=  
item.extra_data['link_to'] %><% end %>
```

```
<!-- end of product description -->
<br /><br />
```

Description:

This displays information about the product that one or more add-ons could be upsold with.

Code:

```
<table><tbody>
  <% seen_upsell_text = Hash.new %>
  <% addon_name_with_to_upsell_text.each_key do |aname_with| %>
    <% a_upsell_with = aname_with.split(/_/)[1] %>
    <% aut_name = aname_with.split(/_/)[0] %>
    <!-- omit addon products that have already been upsold for this item -->
    <% if upsold_name_to_domain.has_key?(aut_name + "_" + item.name) and
upsold_name_to_domain[aut_name + "_" + item.name] == item.name %><%
seen_upsell_text[aname_with] = 1 %><% next %><% end %>
    <% if ((a_upsell_with == 'domain' and product.type.include?('domain'))
or (a_upsell_with == 'package' and product.type.include?('package')) %>
      <% if not seen_upsell_text.has_key?(aname_with) %>
        <% seen_upsell_text[aname_with] = 1 %>
      <!-- start The upsell text -->
      <tr><td colspan='2'><%= addon_name_with_to_upsell_text[aname_with]
%></td></tr>
    <!-- end The upsell text -->
    <% end %>
  <% end %>
```

Description:

This displays the upsell text – the text intended to convince the customer that they should order the extra product.

Code:

```
  <% addon_products.each do |aproduct| %>
    <% if aut_name == aproduct.name and ((a_upsell_with == 'domain' and
product.type.include?('domain')) or (a_upsell_with == 'package' and
product.type.include?('package')) %>
      <% qup = billing.price(aproduct.unit_price) %>
      <% qup_inc_tax = billing.price(aproduct.unit_price, :use_tax_prefs) %>
    <!-- start The Add button -->
    <tr><td><%= aproduct.name %> at <%= billing.currency_symbol.html %><%=
qup_inc_tax %> <%= aproduct.base_renewal_period %></td>
    <td>
      <form action="basket" method="post">
        <input type='hidden' name='add-product' value='<%= aproduct.type
%>:0:base_renewal_period,quoted_unit_price,link_to,exchange_prefix,exchan
ge_domain' />
        <input type='hidden' name='base_renewal_period-0' value='<%=
aproduct.base_renewal_period %>' />
        <input type='hidden' name='quoted_unit_price-0' value='<%= qup %>' />
        <input type='hidden' name='link_to-0' value='<%= item.name.html %>' />
        <input type='hidden' name='stay' value='1' />
        <input type='submit' value='Add to basket' />
      </form>
    </td></tr>
    <!-- end The Add button -->
    <% end %>
  <% end %>
```

Description:

This adds the button so that the customer can add the addon to their basket.

remind-password.rhtml

This page lets the customer request a reminder of their password if they have forgotten it.

Ruby Variables:

Variable	Class:Usage	Value
redirect	String:<%= redirect %>	Nothing or 'login'
from_reg	:<% if from_reg %> Your email address is already registered. <% end %>	Boolean – true if the customer arrived at this page after attempting to register with an email address of an existing account, false otherwise.
security_question	String:<%= security_question %>	The security question text.
username	String:<%= username %>	The email address entered by the customer.

Ruby Code in Template:

Code:

```
<% if redirect %>
<h2>Password Reminder</h2>
<p>Your password has been sent to your registered email
  address, please check your junk mail folder in case it has been filtered
  into it. Please click <a href='<%= redirect %>'>here</a> to return to
  the login page.</p>
<% else %>
```

Description:

If the customer successfully requested a reminder, this code asks them to go to the login page.

Code:

```
<% if from_reg %>
<p>Your email address is already registered. To recover your password,
  please use the form below.</p>
<% end %>
```

Description:

If the customer attempted to register with an email address of an existing account, inform them.

Code:

```
<% if error == :answer_incorrect %>
<p class='error'>Error: your security answer is incorrect</p>
<% elsif error == :bad_username %>
<p class='error'>Error: we can't find that email address</p>
<% elsif error == :no_security_question %>
<p class='error'>Error: we can't find a security question for you. Please
  contact support and give them this error message.</p>
<% end %>
```

Description:

Inform the customer of any errors.

Code:

```
<% if security_question %>
```

```

<tr><th>Security question:</th><td><%= security_question %></td></tr>
<tr><th>Answer:</th><td><input type='password' size='30' name='security-
answer' /></td></tr>
<input type='hidden' name='username' value='<%= username %>'>
<% else %>

```

Description:

If a security question has been configured for this account, ask the customer for the answer.

noprefs.rhtml

This page will be displayed if there is something wrong with your configuration. It should not be modified.

pg-lock.rhtml

This is used during the payment process and should not be modified.

terms.rhtml

This should not be modified.

template-error.rhtml

This is the page that will be displayed if an error was encountered while ruby was processing the current template. You may wish to change the detail about who to contact.

Control Panel pages

The control panel is the page that has a grid of icons and a menu providing access to all the actions available to a logged in user.

Front-page.rhtml

This is the 'home' page of the control panel, and includes the grid of icons and the menu.

Ruby Variables:

Variable	Class:Usage	Value
support_database_allowed?	:<% if support_database_allowed? %> Support Database allowed <% end %>	Boolean – true if support database is allowed, false otherwise.[1]
primary_contact	Object:Welcome back <%= primary_contact.name.first.ht ml %> <% if organisation.name %> at <%= organisation.name.html %><% end %>	This is a contact object. Properties: name.first name.last name.title_short organisation.name organisation.type address.local address.city address.county address.postcode address.country

Variable	Class:Usage	Value
		phone mobile fax email dob

Notes:

[1] The 'support database allowed' flag is off by default and obscurely set by clicking 'Save Changes' in Customer Control Panel below 'Renewal Reminder Intervals'.

Optional code snippets:

i) Welcome

```
<p>Welcome Back, <%= primary_contact.name.title_short %> <%=
primary_contact.name.first.html %> <%= primary_contact.name.last %></p>
```

support-database.rhtml

This page provides the customer with access to the support database. They can select a category or search for a keyword.

Ruby Variables:

Variable	Class:Usage	Value
questions	Object:<% questions.error.each do q %> <a href='?question=<%= q.db_id %>'><%= q.question.html %> <% end %>	This may contain some data following a search or the customer selecting a category. Properties: error – an array of questions config – an array of questions useful – an array of questions
category	Object:<%= category.name.html %>	Properties: name - a string containing the category name. Link - a string containing an HTML <a> tag which links to the category.
top_categories	Array:<% top_categories.each do cat %> <%= cat.name.html %> <% end %>	A collection (array) of category objects.
reseller_strings	Hash:Company <%= reseller_strings.name.html %>	Keys: name – Your Company Name [1] home_web – Home Page URL

Variable	Class:Usage	Value
		[1] support_web – Support Site URL [1] support_email – Support Email Address [1] sales_email – Sales Email Address [1] logo_web – Logo URL[2] cpname – Brand Name [1] cpurl – Control Panel Web Address [2] nsdomain – Virtual Name Server [3] ns1 – derived from nsdomain ns2 – derived from nsdomain

Notes:

[1] These values are configured on Customer Home > Reseller Control Centre > Configure eXtend.

[2] These values are configured on Customer Home > Reseller Control Centre > Domain Control Panel

[3] These values are configured on Customer Home > Reseller Control Centre > Virtual Name Servers

Ruby Code in Template:

Code:

```
<% if questions %>
<h3><% if category %><%= category.name.html %><% else %>Search Results<%
end %></h3>
```

Description:

If there are any results (the customer selected a category or searched for something), either use the category name as the heading or use 'Search Results'.

Code:

```
<% questions.error.each do |q| %>
<li><a href='?question=<%= q.db_id %>'><%= q.question.html %></a></li>
<% end %>
```

Description:

List the questions relating to 'Problems or errors'.

Code:

```
<p>Welcome to the <% if reseller_strings.home_web %>
  <a href="<%= reseller_strings.home_web.html %>"><%=
reseller_strings.name.html %></a>
<% else %>
  <%= reseller_strings.name.html %>
<% end %>
```

Description:

If you have configured home page URL on eXtend Control Panel Configuration, a link to that url will be inserted, otherwise Your Company Name will be used if it has been set.

Code:

```
<% top_categories.each do |cat| %>
```

```

<h3><%= cat.name.html %></h3>
<ul class='cp'>
<% cat.subcategories.each do |scat| %>
<li><%= scat.link %></li>
<% end %>
</ul>
<% end %>

```

Description:

This inserts a set of category headings with lists of subcategories.

support-database-answer.rhtml

This page presents the question and corresponding answer to one of the support questions.

Ruby Variables:

Variable	Class:Usage	Value
printable	:<% unless printable %> Normal <% else %> Suitable for printing <% end %>	Boolean – true if the customer clicked the 'Print Now' link, false otherwise.
question	(Question) Object:<%= question.question.html %>	Question: Properties: question – a String containing the question text. answer – a String containing the answer text. answer_is_html – Boolean .. true if the answer has been flagged as in HTML format, false otherwise. db_id – a Fixnum containing the id of the question.

Ruby Code in Template:

Code:

```

<% unless printable %>
<div class='supportdb-action'>
<a target="_blank" href="?do-print-question=1;question=<%= question.db_id
%>">
Print&nbsp;Now</a>
</div>
<% end %>

```

Description:

Insert the 'Print Now' link if the user has not already followed it.

Code:

```

<% if question.answer_is_html %>
    <%= question.answer %>
<% else %>
    <%= question.answer.html(:maintain_line_breaks) %>

```

<% end %>

Description:

Insert the answer correctly formatted.

Code:

```
<% unless printable %>
<p><a href="javascript:history.go(-1)">Go Back</a></p>
<% else %>
<script type='text/javascript'>
setTimeout("window.print()", 500);
</script>
<% end %>
```

Description:

If customer has not requested a printout, insert a 'Go Back' link, otherwise trigger the printout.

header_printable.rhtml

This is a specialised header for use during printout. It is included in support-database-answer.rhtml and invoices.rhtml.

footer_printable.rhtml

This is a specialised footer for use during printout. It is included in support-database-answer.rhtml and invoices.rhtml.

your-account.rhtml

This page presents a menu enabling the customer to view their invoices, change their account password and update their contact details.

invoices.rhtml

This page either presents a list of invoices associated with the customer or displays an individual invoice if the customer has selected one.

Ruby Variables:

Variable	Class:Usage	Value
invoices	Array:<% invoices.each do inv %> Invoice number <%= inv.invoice_id %> <% end %>	An array of Basket objects.
inv	(Basket) Object:Invoice number <%= inv.invoice_id %>	A single Basket object
invoice_company_name	String:<%= invoice_company_name %>	The company name as configured at Customer Home > Contacts & Billing Details > Contacts & Billing Details, Default contact.
invoice_company_address	String:<%= invoice_company_address %>	The company address as configured at Customer Home > Contacts & Billing Details > Contacts & Billing Details,

Variable	Class:Usage	Value
		Default contact.
billing	(Billing) Object:Tax number <%= billing.tax_number.html %>	Billing object: Properties: currency_symbol – String: the character you selected for 'Currency' in Global Preferences tax_name – String: the tax name you entered in Global Preferences tax_number – String: the tax number you entered in Global Preferences include_tax? - Boolean: true if 'Add Tax to prices' is checked in Global Preferences, false otherwise. tax_applies? - Boolean: true if the 'Tax Rate' is not 0, false otherwise.
customer_org	String: <%= customer_org %>	The organisation name as configured at customise-contact by the customer.
customer_org_type	String:<% if customer_org_type == "company" %> Company <% end %>	The organisation type as configured at customise-contact by the customer.
customer_name	String:<%= customer_name %>	First name joined to Surname by a space (values as configured at customise-contact by the customer).
customer_address	String:<%= customer_address %>	The customer address as configured at customise-contact by the customer.

Ruby Code in Template:

Code:

```
<% if invoices.size == 0 %>
<p>You have no invoices yet</p>
<% else %>
```

Description:

If the customer has no invoices, inform them.

Code:

```
<% invoices.each do |inv| %>
<tr>
<td><a href='?num=<%= inv.invoice_id %>'><%= inv.invoice_id %></a></td>
```

```

<td><%= inv.date %></td>
<td style='text-align: right<% if inv.to_f_withvat < 0 %>; color: red<%
end %>'><%= inv.currency_symbol %><%= sprintf("%.2f",
inv.to_f_withvat.to_money) %></td>
</tr>
<% end %>

```

Description:

Add a row to the table for each invoice, displaying the invoice number and the date on which the items in the invoice were purchased.

Code:

```

<h3>Number <%= inv.invoice_id %></h3>
<p>From: <%= invoice_company_name %><br /><%= invoice_company_address %>
<% if billing.tax_number %>
<br />Tax Registration Number: <%= billing.tax_number.html %>
<% end %>
</p>
<p>To: <% if customer_org_type == "company" %><%= customer_org %><% else
%><%= customer_name %><% end %><br /><%= customer_address %></p>

```

<p>Purchase Date: <%= inv.date %></p>

Description:

Display the following information about the invoice:

Number, the issuer's company name and address, the issuer's tax registration number, the recipient's company name and address, the date of purchase.

Code:

```

<% if inv.discount_total > 0 %>
<tr>
<th colspan='3'>Discount</th>
<td style='text-align: right'><%= inv.currency_symbol %><%=
sprintf("%.2f", if billing.include_tax? then inv.discount_total_withvat
else inv.discount_total end) %></td>
</tr>
<% end %>

```

Description:

If there is any discount, display it.

Code:

```

<% if billing.include_tax? %>
<% else %>
<tr>
<th colspan='3'>Total excluding <%= billing.tax_name %></th>
<td style='text-align: right'><%= inv.currency_symbol %><%=
sprintf("%.2f", inv.to_f) %></td>
</tr>
<tr>
<th colspan='3'><%= billing.tax_name %></th>
<td style='text-align: right'><%= inv.currency_symbol %><%=
sprintf("%.2f", (inv.to_f_withvat - inv.to_f).to_money) %></td>
</tr>
<% end %>

```

Description:

Display the total excluding tax and the total tax amount as part of a 'tax breakdown' if you have disabled 'Add tax to prices'.

Code:

```
<% inv.each do |item| %>
<% next if not(item.product.name) and item.quoted_price == 0 %>
```

Description:

Bypass any items in the invoice that appear to be invalid (have no 'name' and have a price of 0).

Code:

```
<% if item.product.type %>
<% if item.product.type.include?('package') %>Hosting: <%=
item.product.name.html %><br /><% end %>
<% else %>
Refund
<% end %>
```

Description:

If the item is a web hosting package, insert 'Hosting:' followed by the hosting product name e.g. 'Monthly Silver Linux Hosting'. If the item has no type, it is a refund so insert 'Refund'.

Code:

```
<%= item.display_name %><% if item.setup_fee > 0 %>
<br />includes a setup fee of <%= inv.currency_symbol %><%=
sprintf("%.2f",
  if (billing.include_tax?) then
    inv.add_tax(item.setup_fee, inv.tax_percent)
  else
    item.setup_fee
  end
) %>
<% end %>
```

Description:

Insert the display name – for a domain or a web hosting package, this will be the domain name. For an add-on this will be the name of the add-on product. If there is a setup fee associated with this item, display it.

Code:

```
<% if item.product.has_free_products? %>
<br />Includes a free domain
<% end %>
<% if item.discount_percent > 0 %>
<br />Discounted at <%= sprintf("%.2f",item.discount_percent) %>%<br
/>from <%= item.quoted_price_incvat %> to <%=
sprintf("%.2f",item.discounted_quoted_price_incvat) %>
<% end %>
```

Description:

Include details of any free domain or discount.

Code:

```
<td style='text-align: right'><%= inv.currency_symbol %><%=
sprintf("%.2f",
  if (item.discount_percent > 0) then
    if (billing.include_tax?) then
      sprintf("%.2f",item.discounted_quoted_unit_price_incvat.to_f)
    else
      sprintf("%.2f",item.discounted_quoted_unit_price.to_f)
    end
  else
```

```

    if (billing.include_tax?) then
      item.quoted_unit_price_incvat
    else
      item.quoted_unit_price
    end
  end
) %>
</td>

```

Description:

Include the item price for a single unit of the product, with any discount and any tax preferences applied. e.g. if the customer ordered 3 years of a yearly package, show the cost for 1 year of the package.

Code:

```

<td style='text-align: center'><%= item.multiply_renewal_period %></td>
<td style='text-align: right'><%= inv.currency_symbol %><%=
sprintf("%.2f",
  if (item.discount_percent > 0) then
    if (billing.include_tax?) then
      sprintf("%.2f",item.discounted_quoted_price_incvat.to_f)
    else
      sprintf("%.2f",item.discounted_quoted_price.to_f)
    end
  else
    if (billing.include_tax?) then
      sprintf("%.2f",item.quoted_price_incvat)
    else
      sprintf("%.2f",item.quoted_price)
    end
  end
) %>
</td>

```

Description:

Display the 'units' e.g. 3 if the customer paid for 3 years of a yearly web hosting package. Then display the total price e.g. in the above example, the price for 3 years of the package would be shown here.

Code:

```

<% if inv.date == "N/A" %>
<p><strong>Unpaid</strong></p>
<% end %>

```

Description:

If there is no purchase date, then the invoice has not been paid for yet. Inform the customer.

change-password.rhtml

This page enables your customers to change their account password.

Ruby Variables:

Variable	Class:Usage	Value
errors	Array:<% errors.each do e %> <% if e == :mismatch %> Passwords did not match. Please try again. <% end %>	This will contain 0 or more error codes. Possible codes are: mismatch, too_short, facility_disabled,

Variable	Class:Usage	Value
	<% end %>	change_failed
changed	:<% if changed %> You have successfully changed your password. <% end %>	Boolean – true if the password was successfully changed, false otherwise.
change_disabled	:<% if change_disabled %> You cannot change your password. <% end %>	Boolean – true if you have ticked 'Disable Your Control Panel' on Customer Home > Reseller Control Centre > HostPay > Customer Control Panel, false otherwise.

Ruby Code in Template:

Code:

```
<% errors.each do |e| %>
<p class='error'>
<% if e == :mismatch %>
Passwords did not match. Please try again.
<% elsif e == :too_short %>
Your new password is too short. Please enter a longer one.
<% elsif e == :facility_disabled %>
This facility has been disabled.
<% else %>
There was a problem changing your password. Please try again.
<% end %>
```

Description:

If there was an error, inform the customer.

Code:

```
<% if changed %><p class='done'>Password changed!</p><% end %>
```

Description:

On success, inform the customer.

Code:

```
<% if change_disabled %>
<p><input disabled type='submit' value='Change Password' /></p>
<% else %>
```

Description:

Don't allow the customer to apply a change if the configuration says so.

Update-contact.rhtml

The customer can change their contact details on this page.

Ruby Variables:

Variable	Class:Usage	Value
success	:<% if success %> Your contact details have been updated.	Boolean – true if the contact was successfully updated, false otherwise.

Variable	Class:Usage	Value
	<% end %>	
update_disabled	:<% if update_disabled %> You cannot change your contact details. <% end %>	Boolean – true if you have ticked 'Disable Your Control Panel' on Customer Home > Reseller Control Centre > HostPay > Customer Control Panel, false otherwise.

Ruby Code in Template:

Code:

```
<% if success %><p class='done'>Your contact details have been  
updated.</p><% end %>
```

Description:

If the contact was successfully updated, inform the customer.

Code:

```
<% if update_disabled %>  
<p style='clear: both'><input disabled type='submit' value='Update' /></p>  
<% else %>
```

Description:

Don't allow the customer to apply a change if the configuration says so.

services-domains.rhtml

Customers can access their domain management interface via 0 or more links on this page.

Ruby Variables:

Variable	Class:Usage	Value
services	Array:<% services.each do s %> <%= s.name %> <% end %>	Contains an array of ServiceProvision objects. ServiceProvision: Properties: name – String: the service 'name'. For domains and web hosting, this is the domain name, for add-ons this is the associated domain name if there is one, for dedicated- servers this is the domain name of the dedicated server. server_name – String: the domain name associated with the dedicated server e.g. ds- 12345.ds.com server_os – String: the operating system e.g. centos server_ip_address – String: the

Variable	Class:Usage	Value
		ip address server_setup_date – String: the date the server was setup. management_url – String: the url configured at Add-on Products. product_name – String: the name of the product e.g. Monthly Silver Linux Hosting (only available for web hosting products). Product – (Product)Object: a product object item - domain_is_free – for internal use only free_tld – for internal use only linked_domain – String: the domain that an add-on applies to e.g. the domain that Domain Privacy has been associated with.

Ruby Code in Template:

Code:

```
<a target='_blank' href='../service-redirect?service-id=<%= s.db_id %>'>
<%= s.name %></a>
```

Description:

Include an anchor linking to the management page.

services-hosting.rhtml

Customers can access their web hosting management interface via 0 or more links on this page. If you have enabled Hosting Upgrades, customers will be able to order upgrades from this page.

Ruby Variables:

Variable	Class:Usage	Value
allow_upgrades	:<% if allow_upgrades %> Would you like to upgrade? <% end %>	Boolean – true if Allow Hosting Upgrades is ticked in Global Preferences, false otherwise.

Ruby Code in Template:

Code:

```
<a target='_blank' href='../service-redirect?service-id=<%= s.db_id %>'>
<%= s.name %></a>
```

Description:

Include an anchor linking to the management page.

Code:

```

    <% if allow_upgrades %>
        <em><%= s.product_name %></em> (<%=
billing.currency_symbol.html %><%= billing.price(s.product.unit_price,
:use_tax_prefs) %> every <%= s.product.unit_period.to_s_short.html %>)
        <% if s.domain_is_free %>domain renews for
free<% end %>
    <% end %>

```

Description:

If Allow Hosting Upgrades is enabled, include more detail about the current web hosting product.

Code:

```

    <% if allow_upgrades %>
    <dd>
        Upgrade options:
        <form id="upgrade-form-<%= s.db_id %>"
action="<%= url_prefix %>basket" method="post">
        <select id='upgrade-package-<%= s.db_id %>'
name='upgrade-package' onchange='put_upgrade_in_basket(this, <%= s.db_id
%>);'>
            <option value="">No upgrade</option>
<% if (upgrade_products[s.upgrade_key]) then %>
    <% upgrade_products[s.upgrade_key].each do |u| %>
        <% if (service_to_money_owed[s.db_id].nil?) then next end %>
        <% initial_price = billing.price(u.unit_price, :use_tax_prefs)
- billing.price(service_to_money_owed[s.db_id], :use_tax_prefs) +
billing.price(u.setup_fee, :use_tax_prefs) %>
        <% initial_price_s = sprintf('%.2f', initial_price) %>
        <% if (initial_price < 0 ) then next end %>
        <% if (not u.is_a_valid_upgrade_option?(s)) then next end %>
        <% domain_will_be_free = 0 %>
        <% if s.domain_is_free and u.free_products_list %>
            <% free_domain_type = "domain." + s.free_tld %>
            <% u.free_products_list.each do |f| %>
                <% if (f['free_id'] == free_domain_type ) %>
                    <% domain_will_be_free = 1 %>
                <% end %>
            <% end %>
        <% end %>
        <option value='<%= u.type %>-<%= u.base_renewal_period %>:<%=
billing.price(u.unit_price) %>:<%= s.name %>:<%= sprintf("%.2f",
service_to_money_owed[s.db_id]) %>'>
            <%= u.name %>
            <% if u.unit_period.to_s_short %> initially <%=
billing.currency_symbol.html %><%= initial_price_s %> then <%=
billing.currency_symbol.html %><%= billing.price(u.unit_price,
:use_tax_prefs) %> every <%= u.unit_period.to_s_short.html %>
            <% else %>
                <%= billing.currency_symbol.html %><%= initial_price_s
%> (never renews)
            <% end %>
            <% if (s.domain_is_free and domain_will_be_free == 0) %>(Note
- domain will no longer renew for free)<% end %>
        </option>
    <% end %>
<% end %>

```

```
<% end %>
<% end %>
```

```
</select>
</form>
</dd>
<% end %>
```

Description:

If Allow Hosting Upgrades is enabled, present the list of upgrade options. Warn the user if upgrading will result in a domain no longer renewing for free.

services-add-ons.rhtml

Customers can view details of their add-on services on this page.

Ruby Code in Template:

Code:

```
<% if s.management_url and s.management_url.size > 0 %><a target='_blank'
href='<%= s.management_url %>'><% end %>
<%= s.name %><% if s.linked_domain %> for <%= s.linked_domain %><% end %>
purchased on <%= s.date_added.gb_date_from_iso %>
<% if s.management_url and s.management_url.size > 0 %></a><% end %>
```

Description:

Insert details about the add-on – its name, the domain it is associated with (where appropriate), and when it was ordered.

services-dedicated-servers.rhtml

Customers can view details of their dedicated servers and access the management interface on this page.

Ruby Code in Template:

Code:

```
<% services.each do |s| %>
  <tr><td><%= s.server_name %></td><td><%= s.server_os
%></td><td><%= s.server_ip_address %></td><td><%= s.server_setup_date
%></td><td><a target='_blank' href='../service-redirect?service-id=<%=
s.db_id %>'>Now</a></td></tr>
<% end %>
```

Description:

Include a table row for each dedicated server, detailing the domain name, operating system, ip address, setup date and a link to the management interface.

contact-support.rhtml

Your customers can send you support questions or raise / view tickets via this page. If you have specified 'Default Support Form' for Contact Support on Customer Control Panel, this page will trigger an email to be sent to the address you configured when your customer submits a question.

If you specified 'Your own Support Link', your customer will be redirected and this template is not used.

If you specified 'HostPay Ticket System', your customer will be able to view existing tickets and raise new ones on this page.

If you wish to add extra options to the 'Services' drop-down, you can just add extra HTML <option> tags eg:

```
<option value='billing'>Billing</option>
<option value='general'>General</option>
<option value='bespoke'>Bespoke Services</option>
```

Ruby Variables:

Variable	Class:Usage	Value
preference	:<% if preference('support_contact_method') == 'rshop-ticket' %> <% end %>	A string that indicates the configured support method.
tickets	Array:<% tickets.each do ticket_item %> <%= ticket_item["ticket_title"] %> <% end %>	An array of hashes. Each hash has the following keys: ticket_id – internal use only ticket_title – String: the title of the ticket ticket_status – String: one of 'closed', 'open', 'pending-response' last_updated – DateTime: the date and time that the ticket was last updated updates – an array of hashes with keys login_id, update_time, update_text

Ruby Code in Template:

Code:

```
<% if redirect %>
```

```
<h2>Contact Support</h2>
```

```
<p>Your support request has been sent to our team, we will respond as soon as possible. Click here to return to the <a href='<%= redirect %>'>control panel</a>.</p>
```

```
<% else %>
```

Description:

If the customer has submitted a question via email, redirect them to the control panel home page (front_page.rhtml)

Code:

```
<% services.each do |s| %>
```

```
  <option value='<%= s.name.html %>:<%= s.id.html %>'><%=
```

```
s.name.html %><% if preference('support_contact_method') == 'rshop-
ticket' %> (<%= s.service_type.html %>) <% end %></option>
```

```
<% end %>
```

Description:

This offers the customer a list of their services to raise the ticket about.

Code:

```
<% tickets.each do |ticket_item| %>

<tr class="ticketsummary">

<td><p onmouseout="this.style.cursor='default';"
onmouseover="this.style.cursor='pointer';" onclick="toggle_div('div-item-
<%= ticket_item["ticket_id"] %>')" style="cursor: pointer;">

<u><%= ticket_item["ticket_id"] %></u></p>

</td>

<td><p onmouseout="this.style.cursor='default';"
onmouseover="this.style.cursor='pointer';" onclick="toggle_div('div-item-
<%= ticket_item["ticket_id"] %>')" style="cursor: pointer;">

<u><%= ticket_item["ticket_title"] %></u></p></td>

<td><p onmouseout="this.style.cursor='default';"
onmouseover="this.style.cursor='pointer';" onclick="toggle_div('div-item-
<%= ticket_item["ticket_id"] %>')" style="cursor: pointer;">.png"
alt="<%= ticket_item["ticket_status"].capitalize %>" />

<u><%= ticket_item["ticket_status"].capitalize %></u></p></td>

<td><p onmouseout="this.style.cursor='default';"
onmouseover="this.style.cursor='pointer';" onclick="toggle_div('div-item-
<%= ticket_item["ticket_id"] %>')" style="cursor: pointer;">

<u><%= ticket_item["last_updated"].to_s.gsub(/^(\\d{4})\\-(\\d{2})\\-(\\d{2})
(\\d{2})\\:(\\d{2})\\:(\\d{2})$/, '\\3/\\2/\\1 \\4:\\5') %></u></p></td>

</tr>
```

Description:

Insert information about each existing ticket: reference number, title, status, date and time of last update.

Code:

```
<% ticket_item["updates"].each do |ticket_update| %>

<tr style="height:0%">

<td><% if ticket_update["login_id"] %>You<% else %>Support<% end %>

</td>

<td><%= ticket_update["update_time"].to_s.gsub(/^(\\d{4})\\-(\\d{2})\\-(\\d{2})
(\\d{2})\\:(\\d{2})\\:(\\d{2})$/, '\\3/\\2/\\1 \\4:\\5') %>
```

</td>

<td><%= ticket_update["update_text"].to_s.html.gsub(/\n/, "
") %>

</td>

</tr>

<% end %>

Description:

Include information about any previous updates.

Renew.rhtml

If your customer has services that are due to be renewed, they can view them on this page and choose whether to renew or cancel the service. There will also be a list of services that are not currently due for renewal.

Ruby Variables:

Variable	Class:Usage	Value
renewals	Array:<% renewals.each do renewal %> Item will expire on <%= renewal['abort_date'].to_s.gb_date_from_iso %> <% end %>	An array of hashes representing services that are due for renewal with the following keys: product – a Product object is_a_free_product – Fixnum: 1 if the product is free (costs nothing), 0 otherwise. quoted_unit_price – Float: the price as it was when the service was ordered for 1 unit of the product. secondary_id – String: the domain name abort_date – Date: the date on which the service will expire. provision_id – Fixnum: an internal reference. Period – String: the renewal period e.g. '2 years' multiplier – Fixnum: the number of 'units' of the product.
undue_renewals	Array:<% undue_renewals.each do renewal %> Item will expire on <%= renewal['abort_date'].to_s.gb_date_from_iso %> <% end %>	An array of hashes representing services that are not due for renewal. See renewals above.

Variable	Class:Usage	Value
cancel_email_sent	:<% if cancel_email_sent %> Thank you your email has been sent <% end %>	Boolean – true if an email has been sent to you, notifying you that your customer wishes to cancel a service, false otherwise.
freedomain_ds_link	Hash:<% if (freedomain_ds_link.has_key?(renewal['secondary_id'])) %> If you cancel the dedicated server, this domain will no longer renew for free. <% end %>	Hash with domain names as keys and references to dedicated server data as values.

Ruby Code in Template:

Code:

```
<% if cancel_email_sent %>
```

```
<h3>Thank You</h3>
```

```
<p>
```

```
Your request has been sent.
```

```
</p>
```

```
<% end %>
```

Description:

If the user requested cancellation, inform them that the email has been sent.

Code:

```
<% renewals.each do |renewal| %>
```

```
<% product = renewal['product'] %>
```

```
<tr>
```

```
<!-- Service description -->
```

```
<td><%= product.name %>: <%= renewal['secondary_id'] %>
```

```
    <% if product.has_free_products? %>
```

```
        <br />includes free domains
```

```
        <% product.free_products_list.each do | free_product |
```

```
%>
```

```
            <%= free_product["free_name"] %>
```

```
        <% end %>
```

```
<% end %>
```

```

        <% if renewal['is_a_free_product'] == 1 and
renewal['quoted_unit_price'] == 0 %>

            <% if
(freedomain_ds_link.has_key?(renewal['secondary_id'])) %>

                <br />If you cancel Dedicated Server <%=
ds_allowance_id_to_name[freedomain_ds_link[renewal['secondary_id']]] %>
the price of this will change.

            <% else %>

                <br />If you cancel Web Hosting for <%=
renewal['secondary_id'] %> the price of this will change.

            <% end %>

        <% end %>

```

</td>

Description:

Insert the service description. The description will be of the format e.g.:
Silver Linux Hosting: somedomain.biz
includes free domains Global Domain (biz)
or
UK Domain (co.uk): anotherdomain.co.uk
If you cancel Web Hosting for anotherdomain.co.uk the price of this will change.

Code:

```
<!-- Expiry date -->
```

```
<td><%= renewal['abort_date'].to_s.gb_date_from_iso %></td>
```

Description:

Include the expiry date.

Code:

```
<!-- Renewal Period -->
```

<td>

```

        <% if product.variable_period? %>

            <% term_count = product.unit_period.count %>

            <select id='multiply_renewal_period-<%=
renewal['provision_id'] %>' name='multiply_renewal_period-<%=
renewal['provision_id'] %>' onchange='javascript:update_total(<%=
renewal['provision_id'] %>, value)''>

                <% (product.min_years ..
product.variable_period_max).each do |multiplier| %>

                    <% multiplied_term = term_count * multiplier %>

                    <% break if multiplied_term >9 %>

```

```

                <option value='<%= multiplier %>' <% if
product.default_period == multiplier %>selected='selected'<% end %>><%=
product.unit_period.to_s(multiplier) %></option>

```

```

                <% end %>

```

```

            </select>

```

```

        <% else %>

```

```

                <%= renewal['period'].to_s %>

```

```

        <% end %>

```

```

        <input type='hidden' id='unit-price-<%= renewal['provision_id']
%>' value='<%= sprintf("%.2f",
billing.price(renewal['quoted_unit_price'], :use_tax_prefs)) %>' />

```

```

    </td>

```

Description:

Include the renewal period. If appropriate allow the customer to select a renewal period.

Code:

```

<!-- Price -->

```

```

<td><%= billing.currency_symbol.html %><span id='price-<%=
renewal['provision_id'] %>'><%= sprintf("%.2f",
renewal['multiplier'].to_f * billing.price(renewal['quoted_unit_price'],
:use_tax_prefs)) %></span>

```

```

</td>

```

Description:

Include the price of the renewal.

Code:

```

<!-- Action selector - Renew or Cancel -->

```

```

<td><% if not product.name.match(/Invalid/) %><select name='renewal:<%=
renewal['provision_id'] %>' onchange="if(this.value == 'cancel') {
document.getElementById('cancellation_container').style.display = ''; }">

```

```

<option>-- Please select --</option>

```

```

<option value='renew'>Renew</option>

```

```

<option value='cancel'>Cancel</option>

```

```

</select><% else %><a href="<%= url_prefix %>contact-support">Contact
Support</a><% end %></td>

```

```

</tr>

```

```

<% end %>

```

Description:

If the product is invalid, ask the customer to contact support, otherwise let them choose to renew

or cancel. The product can be invalid if you have deleted it since the customer placed their original order.

renew-payment.rhtml

This is very similar to payment-options.rhtml and it enables your customers to pay for their renewals.

set-security-question.rhtml

If your customer's account does not have a specified security question and corresponding answer they will be taken to this page following successful login.

Ruby Code in Template:

Code:

```
<%= field('dob-d') %> / <%= field('dob-m') %> / <%= field('dob-y') %>
```

Description:

Include the day, month, and year fields for the customer's date of birth.

Code:

```
<% if field('source') then %>
```

```
<h3>Other Information</h3>
```

```
<p>Where did you hear about us? <%= field('source') %> </p>
```

```
<% end %>
```

Description:

Include the 'where did you hear about us' data.

payment-verification.rhtml

This page is used during the 3Dsecure payment process for PayPal Website Payments Pro Direct payment method.

payment-authenticated.rhtml

This page is used during the 3Dsecure payment process for PayPal Website Payments Pro Direct payment method.

payment-notenrolled.rhtml

This page is used during the 3Dsecure payment process for PayPal Website Payments Pro Direct payment method.

payment-verified.rhtml

This page is used during the 3Dsecure payment process for PayPal Website Payments Pro Direct payment method.

payment-notify.rhtml

This page may be used during the payment process. It may be triggered by an incoming payment notification, but will not normally be seen by your customers, and therefore probably should not be modified.

feedback.rhtml

You can offer your customers the opportunity to suggest improvements or new features on this page by posing questions. Their answers will be emailed to you and they will receive a thank you email.

Ruby Variables:

Variable	Class:Usage	Value
feedback_questions	Array: <% feedback_questions each do question %> Question: <%= question.text %> Answer: <textarea rows='4' cols='40'>	An array of hashes containing the question text and a flag indicating whether the question should be rated or not.

Ruby Code in Template:

Code:

```
<% i = 0 %>
<% feedback_questions.each do |question| %>
<p><%= question['text'] %></p>
<% if question['is_rated'] == 1 %>
<select name='<%= i %>'>
<option value='Very Good'>Very Good</option>
<option value='Good'>Good</option>
<option value='Average'>Average</option>
<option value='Poor'>Poor</option>
<option value='Very Poor'>Very Poor</option>
</select>
<% else %>
<textarea name='<%= i %>' cols='40' rows='5'></textarea>
<% end %>
<% i=i+1 %>
<% end %>
```

Description:

Insert each question into the page. If the question is rated add a select with the different rating options, otherwise add a textarea into which the customer can type their answer.

invoice_header_printable.rhtml

You can insert extra content e.g. your company logo at the top of the printable invoice by modifying this template.

invoice_footer_printable.rhtml

You can insert extra content e.g. your company address at the bottom of the printable invoice by modifying this template.

Ruby variable availability

This table lists in which non-static pages each ruby variable can be used. Note that no variables can be used in any static page.

Variable	Pages
url_prefix	all
logged_in?	all
username	All, but only has a value once the user has logged in.
domain_types	index.rhtml dac.rhtml dac-results.rhtml domain-needed.rhtml
user_menu	All, but only has a value once the user has logged in.
ds_products_for_sale?	all
domain_registration_availability	dac-results.rhtml
domain_transfer_availability	dac-results.rhtml
domains	package-chooser.rhtml add-ons.rhtml
hosting_products	package-chooser.rhtml
domain_references_allowed?	all
error_message	Domain-needed.rhtml basket.rhtml customise-contact.rhtml
addon_products	Add-ons.rhtml addon-upsell.rhtml
basket	Add-ons.rhtml basket.rhtml addon-upsell.rhtml
has_basket_items?	basket.rhtml
subdomains	Add-ons.rhtml
error	Register.rhtml remind-password.rhtml set-security-question.rhtml
paypalpro_page_style	payment-options.rhtml renew-payment.rhtml
gateways	payment-options.rhtml renew-payment.rhtml
freegateway	payment-options.rhtml

Variable	Pages
	renew-payment.rhtml
dropdown_gateways	payment-options.rhtml renew-payment.rhtml
invoice_number	Result-complete.rhtml
address	Result-cheque.rhtml
payable	Result-cheque.rhtml
amount	Result-cheque.rhtml
invoice_instructions	Result-invoice.rhtml
Invoice_email	Result-invoice.rhtml
login_attempted	Login-reminder.rhtml
target_username	Login-reminder.rhtml
discount_code	Login-reminder.rhtml
dual_core_for_sale	dedicated-servers.rhtml
quad_core_for_sale	dedicated-servers.rhtml
ds_spec_details	dedicated-servers.rhtml
os_to_name	dedicated-servers.rhtml
os_list	dedicated-servers.rhtml
default_cpanel_price	dedicated-servers.rhtml
default_backup_price	dedicated-servers.rhtml
free_products	dedicated-servers.rhtml
upsellable_item_list	addon-upsell.rhtml
addon_name_with_to_upsell_text	addon-upsell.rhtml
upsold_name_to_domain	addon-upsell.rhtml
support_database_allowed?	Front-page.rhtml
primary_contact	Front-page.rhtml
redirect	remind-password.rhtml contact-support.rhtml
from_reg	remind-password.rhtml
security_question	remind-password.rhtml
questions	support-database.rhtml
category	support-database.rhtml
top_categories	support-database.rhtml
reseller_strings	support-database.rhtml
printable	Support-database-answer.rhtml
question	Support-database-answer.rhtml

Variable	Pages
Invoices	Invoices.rhtml
inv	invoices.rhtml
invoice_company_name	invoices.rhtml
invoice_company_address	invoices.rhtml
billing	all
customer_org	invoices.rhtml
customer_org_type	invoices.rhtml
customer_name	invoices.rhtml
customer_address	invoices.rhtml
errors	change-password.rhtml
changed	change-password.rhtml
change_disabled	change-password.rhtml
success	update-contact.rhtml
update_disabled	update-contact.rhtml
services	services-domains.rhtml services-hosting.rhtml services-add-ons.rhtml services-dedicated-servers.rhtml
allow_upgrades	Services-hosting.rhtml
preference	contact-support.rhtml
tickets	contact-support.rhtml
renewals	renew.rhtml
undue_renewals	renew.rhtml
cancel_email_sent	renew.rhtml
freedomain_ds_link	renew.rhtml