

Foundation Store

Table of Contents

<u>1. The Foundation Store</u>	1
<u>2. Tree design</u>	3
<u>3. The Catalog Directory</u>	5
<u>4. Page Templates</u>	7
<u>4.1. Template File Locations</u>	7
<u>4.2. Themes</u>	10
<u>4.3. Template Definition Files</u>	11
<u>4.4. Edit Page Function</u>	15
<u>4.5. Region Templates</u>	16
<u>4.6. Template Page List</u>	20
<u>4.7. Special Page List</u>	21
<u>4.8. Components</u>	22
<u>5. The Database Tables</u>	33
<u>5.1. 2ndDayAir.csv</u>	34
<u>5.2. 450.csv</u>	34
<u>5.3. Ground.csv</u>	34
<u>5.4. NextDayAir.csv</u>	34
<u>5.5. access.asc</u>	34
<u>5.6. affiliate</u>	38
<u>5.7. area</u>	40
<u>5.8. banner</u>	45
<u>5.9. cat</u>	46
<u>5.10. country</u>	50
<u>5.11. downloadable</u>	51
<u>5.12. files.txt</u>	52
<u>5.13. gift_certs.txt</u>	52
<u>5.14. inventory.txt</u>	52
<u>5.15. locale.txt</u>	52
<u>5.16. merchandising.txt</u>	52
<u>5.17. mv_metadata</u>	53
<u>5.18. options</u>	53
<u>5.19. order_returns.txt</u>	60
<u>5.20. orderline.txt</u>	60
<u>5.21. pricing</u>	61
<u>5.22. products</u>	63
<u>5.23. products.category.txt</u>	67
<u>5.24. route.txt</u>	68
<u>5.25. saletax.asc</u>	68
<u>5.26. shipping.asc</u>	68
<u>5.27. state.txt</u>	68
<u>5.28. transactions.txt</u>	69
<u>5.29. userdb.txt</u>	70
<u>5.30. variable.txt</u>	71

Table of Contents

<u>6. HTML Hypertext links</u>	73
<u>7. Images</u>	75
<u>8. Browser Cookies</u>	77
<u>9. Dependencies in administration</u>	79

1. The Foundation Store

The Foundation store is distributed with Interchange to give you a starting point with which to build your e-business. While the Foundation store is designed to be relatively easy to start with, it is still a full-featured demonstration of a number of Interchange capabilities. Once you understand the Foundation store and how it works you are well on your way to understanding the Interchange software.

The following is a list of some popular features:

Category Searches

Regardless of the number of products in a catalog, categorizing them makes them easier to find. Pick a field in the database, typically named `category`, and classify the products for search using Interchange.

Images

You can display a thumbnail image for the items that have images. To do this, add an image field in the database. (See the 'image' field of the products database.)

Related Items

You can embed searches of similar products on an individual product display page with the `[query . . .]` or `[loop . . .]` tags. Or, if customer data is developed, search a past order database and display products that would be of interest to that customer.

Reviews/Testimonials

You can key the placement of a review or testimonial on the existence of a file being in a certain directory. This is reasonable to do when a user is viewing a single product.

2. Tree design

By determining how users will enter and exit the catalog, complex and intelligent conditional schemes are possible, especially if the Cookies capability is exercised. However, it is recommended that simplicity be used. Consumers will not make purchases if they can't navigate their way around the catalog.

It is important to remember that users will lose their session (and items in their shopping cart) if their browser does not accept cookies and they leave the site. Interchange addresses this problem by using the `area` and `page` tags. If you are using frames, source all frame panes containing Interchange links from an initial page served by Interchange. If you don't do this, the user may have multiple session IDs depending on which frame generated the link.

Note that Interchange can work properly even if the browser doesn't store cookies. In this situation Interchange inserts a session ID into each URL; if the ID is preserved as the user navigates from page to page the session will remain intact.

3. The Catalog Directory

Interchange pages are contained in the catalog directory. Each individual catalog has its own base directory. The catalog directory has the following structure by default:

catalog.cfg

File containing configuration directives for a particular catalog. Configuration settings established in the `catalog.cfg` directory will not effect any other catalogs running under the version of Interchange you are using. Subcatalogs can have differing information in a file named for that subcatalog.

config

Directory that will be read when directives are set with the `filename` notation. For example, the file `config/static.pages` will be read when the following directive is encountered in the `catalog.cfg` file.

```
StaticPage <static.pages
```

This directory also contains template information used with the `makecat` program.

error.log

File which contains catalog-specific errors. It is also where any syntax errors in embedded Perl code are shown.

etc

Directory normally used for tracking files, order profiles, and other configuration and log information.

pages

Directory that contains the pages of the catalog. This can be considered to be the "document root" of the catalog. Pages contained therein are called with the path information after the script name. For example:

```
/cgi-bin/simple/products/gold will call the page in the file  
pages/products/gold.html.
```

products

Directory that contains database source files, including the special Interchange databases `shipping.asc`, `pricing.asc` (and other shipping database files).

session

Directory that contains session files.

tmp

The temporary or scratch directory used for various storage reasons, like retired ID numbers, search paging

Foundation Store

files, sort tests, import temporary files, etc. This is the default set by ScratchDir. It can be redefined to be located on another partition.

4. Page Templates

This section describes the files located in the Foundation demo.

4.1. Template File Locations

This diagram shows the directory and file structure used for the default Foundation 'templates' directory. The base will be a directory with the name of your catalog, here called CATROOT.

```
CATROOT/
|
|----templates/
|   |----cart
|   |----components/
|   |   |----affiliate_receptor
|   |   |----best_horizontal
|   |   |----best_vertical
|   |   |----cart
|   |   |----cart_display
|   |   |----cart_tiny
|   |   |----category_vertical
|   |   |----cross_horizontal
|   |   |----cross_vertical
|   |   |----modular_buy
|   |   |----modular_update
|   |   |----none
|   |   |----promo
|   |   |----promo_horizontal
|   |   |----promo_vertical
|   |   |----random
|   |   |----random_horizontal
|   |   |----random_vertical
|   |   |----saved_carts_list_small
|   |   |----search_box_small
|   |   |----upsell
|   |   |----upsell_horizontal
|   |   |----upsell_vertical
|   |----default --> foundation
|   |----foundation/
|   |   |----cart
|   |   |----fullwidth
|   |   |----leftonly
|   |   |----leftright
|   |   |----regions/
|   |   |   |----LEFTONLY_BOTTOM
|   |   |   |----LEFTONLY_TOP
|   |   |   |----LEFTRIGHT_BOTTOM
|   |   |   |----LEFTRIGHT_TOP
|   |   |   |----NOLEFT_BOTTOM
|   |   |   |----NOLEFT_TOP
|   |   |----simple
|   |   |----theme.cfg
|   |----fullwidth
|   |----leftonly
|   |----leftright
|   |----regions/
|   |   |----LEFTONLY_BOTTOM
|   |   |----LEFTONLY_TOP
```

Foundation Store

```
|----LEFTRIGHT_BOTTOM
|----LEFTRIGHT_TOP
|----NOLEFT_BOTTOM
|----NOLEFT_TOP
|----sampledata/
|----computers/
  |----images/
    |----items/
      |----generic.gif
      |----gift_certificate_large.gif
      |----yourimage.gif
    |----thumb/
      |----generic_thumb.gif
      |----gift_certificate.gif
      |----thumb.gif
  |----products/
    |----inventory.txt
    |----merchandising.txt
    |----mv_metadata.asc
    |----options.txt
    |----pricing.txt
    |----products.txt
    |----userdb.txt
|----reports/
  |----download/
    |----00352as.pdf
    |----11993ab.pdf
    |----22083da.pdf
    |----49503cg.pdf
    |----59330rt.pdf
    |----59402fw.pdf
    |----66548ch.pdf
    |----73358ee.pdf
    |----83491vp.pdf
    |----90773sh.pdf
  |----products/
    |----mv_metadata.asc
    |----products.txt
    |----userdb.txt
|----tools/
  |----etc/
    |----after.cfg
    |----before.cfg
  |----images/
    |----items/
      |----os28004.gif
      |----os28005.gif
      |----os28006.gif
      |----os28007.gif
      |----os28008.gif
      |----os28009.gif
      |----os28011.gif
      |----os28044.gif
      |----os28057a.gif
      |----os28057b.gif
      |----os28057c.gif
      |----os28062.gif
      |----os28064.gif
      |----os28065.gif
      |----os28066.gif
      |----os28068.gif
      |----os28068a.gif
```

Foundation Store

```
|----os28068b.gif
|----os28069.gif
|----os28070.gif
|----os28072.gif
|----os28073.gif
|----os28074.gif
|----os28075.gif
|----os28076.gif
|----os28077.gif
|----os28080.gif
|----os28081.gif
|----os28082.gif
|----os28084.gif
|----os28085.gif
|----os28086.gif
|----os28087.gif
|----os28108.gif
|----os28109.gif
|----os28110.gif
|----os28111.gif
|----os28112.gif
|----os28113.gif
|----os29000.gif
|----thumb/
|----gift_certificate.gif
|----os28004_b.gif
|----os28005_b.gif
|----os28006_b.gif
|----os28007_b.gif
|----os28008_b.gif
|----os28009_b.gif
|----os28011_b.gif
|----os28044_b.gif
|----os28057a_b.gif
|----os28057b_b.gif
|----os28057c_b.gif
|----os28062_b.gif
|----os28064_b.gif
|----os28065_b.gif
|----os28066_b.gif
|----os28068_b.gif
|----os28068a_b.gif
|----os28068b_b.gif
|----os28069_b.gif
|----os28070_b.gif
|----os28072_b.gif
|----os28073_b.gif
|----os28074_b.gif
|----os28075_b.gif
|----os28076_b.gif
|----os28077_b.gif
|----os28080_b.gif
|----os28081_b.gif
|----os28082_b.gif
|----os28084_b.gif
|----os28085_b.gif
|----os28086_b.gif
|----os28087_b.gif
|----os28108_b.gif
|----os28109_b.gif
|----os28110_b.gif
|----os28111_b.gif
```

```

        |----os28112_b.gif
        |----os28113_b.gif
        |----os29000_b.gif
|----products/
    |----affiliate.txt
    |----area.txt
    |----cat.txt
    |----inventory.txt
    |----merchandising.txt
    |----mv_metadata.asc
    |----options.txt
    |----orderline.txt
    |----pricing.txt
    |----products.txt
    |----transactions.txt
    |----userdb.txt

```

4.2. Themes

This section explains how themes are defined in Interchange via the `STYLE` variable and the theme configuration file, `theme.cfg`.

4.2.1. STYLE

The `STYLE` variable in `CATROOT/products/variable.txt` indicates the template style to be used as the theme for the catalog; the appropriate templates for that theme are found in `CATROOT/templates/___STYLE___/`. (To change the value of the `STYLE` variable, either edit `variable.txt` directly or use the table editor feature of the admin interface.)

The default theme for Interchange is the Foundation demo; hence, the `STYLE` variable is assigned the value 'Foundation' in `variable.txt`. The theme is defined in `catalog.cfg` as follows (line numbers added):

```

# Here we set up the catalog theme.

1 ParseVariables Yes

2 ifndef STYLE
3 Variable STYLE default
4 endif
5 include templates/___STYLE___/theme.cfg

```

Variables that make up the look and feel of the `STYLE` (theme) are defined in the file `CATROOT/templates/foundation/theme.cfg`, which is read by Interchange in line 5 above.

4.2.2. theme.cfg

The file `CATROOT/templates/foundation/theme.cfg` serves three purposes:

1. It defines the `THEME` and `THEME_IMG_DIR` variables,
2. It defines a cascading style sheet for the theme, and
3. It defines the location of region templates according to the traffic settings for the Interchange daemon.

The `THEME` variable is used to set the location of the region templates in the traffic settings section of the `theme.cfg` file. It is also used in the cart template definition file (`CATROOT/templates/cart`) to set the path of

an image. The `THEME_IMG_DIR` variable is used to set image paths in the template region files and the template component files.

The look and feel of the Foundation theme are defined primarily in the cascading style sheet specified in the `theme.cfg` file. This

The Interchange `TRAFFIC` setting, defined system-wide in `interchange.cfg`, is described in the `??document??`. The settings in `theme.cfg` pertain to the location of region templates for the high and low traffic settings. For example, if you need to define a separate set of high traffic templates, you would change the `ConfigDir` variable in `theme.cfg` to point to the directory containing those templates.

4.3. Template Definition Files

The template definition files store the name and description of the template as well as components and options for that template.

```
templates/cart
templates/fullwidth
templates/leftonly
templates/leftright

templates/foundation/cart
templates/foundation/fullwidth
templates/foundation/leftonly
templates/foundation/leftright
templates/foundation/simple
```

4.3.1. Template Walkthrough -- leftonly

This section is best read while viewing the file `CATROOT/templates/leftonly` and the 'Edit Page' page in the Content Editor of the Interchange Administration Tool.

Looking at the example template definition file, all lines located between the `[comment]` and `[/comment]` tags (lines 1 and 53) control what is available in the Edit Page screen of the Administration Tool.

Lines 2–5: Template specification

```
2 ui_template: Yes
3 ui_template_name: leftonly
4 ui_template_layout: LEFTONLY_TOP, UI_CONTENT, LEFTONLY_BOTTOM
5 ui_template_description: Page with top/left areas.
```

Line 2 indicates that this file is a template for the user interface. Line 3 names the template, while Line 4 indicates the regions that comprise the template and that will eventually make up the new page that is created from the template. Line 5 provides a description used to identify the template when it appears in a Select Template pull-down menu on the Edit Page of the Administration Tool. This description can be changed or modified to better describe a new template or a template that is created from the stock templates provided with Interchange.

Lines 7–8: Break

```
7 break:
8         widget: break
```

This code creates a separation line in the Edit Page between sets of options. In the default Interchange installation the line is grey, but the color can be changed. Note — Changing this color applies the change to any catalog served by Interchange.

Lines 10–11: Page Title

```
10 page_title:
11     description: Page title
```

This code tells Interchange to display a text field on the Edit Page for entering the page title ('Title of New Page' in this example). The value entered is assigned to the scratch variable `page_title` and is set as a default value at the bottom of the template definition file using the following ITL:

```
54 [set page_title][set]
```

which, in turn, sets the scratch variable on the new page using the tag

```
[set page_title]Title of New Page[set]
```

The scratch variable `page_title` is parsed by the following code in the region template specified in the template definition file and called in the new page:

```
<title>[scratch page_title]</title>
```

Lines 13–15: Page Banner

```
13 page_banner:
14     description: Page banner
15     help: Defaults to page title
```

Assigns a textual title for the page to the scratch variable `page_banner`, which is assigned by the following ITL:

```
55 [set page_banner][set]
```

The scratch variable `page_banner` is set on the new page using the tag

```
[set page_banner]Banner of New Page[set]
```

The scratch variable can be parsed in the region template by this code:

```
[either]
  [scratch page_banner]
[or]
  [scratch page_title]
[/either]
```

This results in the page banner being displayed if defined. Otherwise, the page title is used.

Lines 17–20: Members Only

```
17 members_only:
18     options: 1=Yes,0=No*
19     widget: radio
```

```
20      description: Members only
```

This creates a radio-button form element on the Edit Page with the user can specify whether the page can be accessed if a visitor is logged in (Yes) or not (No). The default is indicated by an asterisk.

The scratch variable `members_only` is assigned by the ITL code

```
56 [set members_only][set]
```

and set on the new page using the tag

```
[set members_only]0[/set]
```

if the page can be accessed without logging in or

```
[set members_only]1[/set]
```

if it can not.

The `members_only` function is handled by the following code within each region template file:

```
[if scratch members_only]
  [set members_only][set]
  [if !session logged_in]
    [set mv_successpage]@MV_PAGE@[set]
    [bounce page=login]
  [/if]
[/if]
```

This code says that if "members only" is set to yes, and the visitor is logged in, to display the page. Otherwise, redirect the visitor to the login page.

Lines 22–23: Break

```
22 break1:
23      widget: break
```

Another separation line.

Lines 25–28: Horizontal Before Component

```
25 component_before:
26      options: =none, best_horizontal=Best Sellers, cross_horizontal=Cross sell, \
      promo_horizontal=Promotion, random_horizontal=Random items, \
      upsell_horizontal=Upsell
27      widget: select
28      description: Component before content
```

This allows the inclusion of a defined component (included in the `CATROOT/templates/components` directory) to be displayed before, or above, the page's content. It provides a pull-down menu on the Edit Page displaying the available components. The components, identified here on line 26, can be assigned a name via the `value=name` convention.

The scratch variable `component_before` is assigned in the template definition file by the ITL code

Foundation Store

```
57 [set component_before][set]
```

It is called with the following code within the LEFTRIGHT_TOP, LEFTONLY_TOP, and NOLEFT_TOP region templates:

```
[if scratch component_before]
  [include file="templates/components/[scratch component_before]"]
[/if]
```

Lines 30–33: Horizontal After Component

```
30 component_after:
31     options: =none, best_horizontal=Best Sellers, cross_horizontal=Cross sell, \
    promo_horizontal=Promotion, random_horizontal=Random items, \
    upsell_horizontal=Upsell
32     widget: select
33     description: Component after content
```

Similar to component_before, this allows the inclusion of a defined component after, or below, the page's content.

The scratch variable component_before is assigned in the template definition file by the ITL code

```
58 [set component_after][set]
```

It is called with the following code within the LEFTRIGHT_BOTTOM and LEFTONLY_BOTTOM region templates:

```
[if scratch component_after]
  [include file="templates/components/[scratch component_after]"]
[/if]
```

Lines 35–38: Horizontal Item Width

```
35 component_hsize:
36     options: 1,2,3*
37     widget: select
38     description: Component items horizontal
```

This setting allows you to choose how many items the horizontal components display. For example, the horizontal best sellers component ("best_horizontal") uses this setting to randomly select the best sellers. Notice the default is 3 if nothing is defined. It is called by the following code in the promo_horizontal and random_horizontal components in the Foundation demo.

```
random="[either][scratch component_hsize][or]2[/either]"
```

Lines 40–45: Before/After Banner

```
40 hbanner:
41     options: =--custom--, Also see..., Best Sellers, New items, \
    Some of our fine products, Specials, You might also like
42     widget: move_combo
43     width: 40
44     description: Before/after Banner
45     help: Banner for Before/after component
```

Allows a title for the horizontal components to be defined to be displayed in a header above the component's items. It is called with the [scratch hbanner] tag and used in the Foundation demo in the random_horizontal component.

Lines 47–51: Special Tag

```
47 hpromo_type:
48     options: specials=Specials, new=New items
49     widget: select
50     description: Special tag
51     help: Only for a horizontal Promotion
```

This setting is only viable when a promotion is used for a horizontal component. It tells the promotional component which row(s) to evaluate in the merchandising table for display within the component. This setting, used in the promo_horizontal component, typically correlates to the featured column of the merchandising table as follows:

```
[query arrayref=main
  sql="
    SELECT sku,timed_promotion,start_date,finish_date
    FROM merchandising
    WHERE featured = '[scratch hpromo_type]'
  "
[/query]
```

4.4. Edit Page Function

Creating a page with the following specifications using the Edit Page function results in the HTML and ITL code displayed below.

Specifications:

```
Template:                Page with top/left areas.
Page title:              test
Page banner:             test
Members only:            No
Component before content: Best Sellers
Component after content: Random items
Component items horizontal: 3
Before/after Banner:    New items
Special tag:              Specials
Content:                  <P>My first HTML/ITL page!
```

Resulting code:

```
[comment]
ui_template: Yes
ui_template_name: leftonly
[/comment]

[set hbanner]New items[/set]
[set page_title]test[/set]
[set hpromo_type]specials[/set]
[set component_hsize]3[/set]
[set page_banner]test[/set]
[set members_only]0[/set]
```

```
[set component_before]best_horizontal[/set]
[set component_after]random_horizontal[/set]
@_LEFTONLY_TOP_@

<!-- BEGIN CONTENT -->
<P>My first HTML/ITL page!
<!-- END CONTENT -->

@_LEFTONLY_BOTTOM_@
```

An important point demonstrated here is the inclusion of the region templates LEFTONLY_TOP and LEFTONLY_BOTTOM through the @_VARIABLE_NAME_@ notation. These are included because of line 4 of the leftonly template definition file:

```
4 ui_template_layout: LEFTONLY_TOP, UI_CONTENT, LEFTONLY_BOTTOM
```

However, understand that you are free to change the region templates used in the file by editing the file itself or, better yet, using an existing region as a starting point for a region of your own design.

The next section explains the structure of region templates.

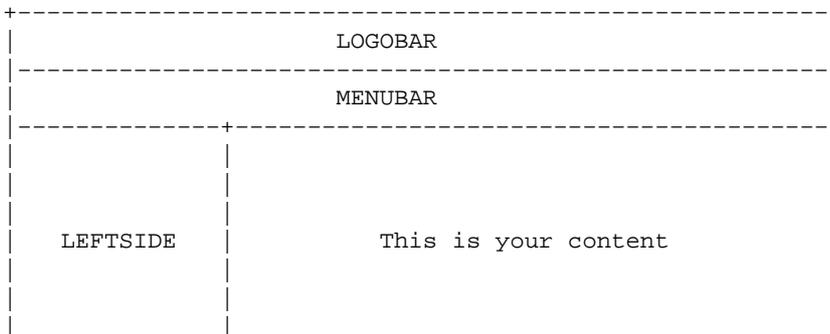
4.5. Region Templates

Interchange region templates (or "regions") are portions of HTML and ITL that are included in pages within a catalog. Using regions, along with the cascading style sheet defined in theme.cfg, allows you to control the look and feel of specific parts of each catalog page.

The default Foundation region set, found in CATROOT/templates/foundation/regions, includes the following:

```
LEFTONLY_TOP
LEFTONLY_BOTTOM
LEFTRIGHT_TOP
LEFTRIGHT_BOTTOM
NOLEFT_TOP
NOLEFT_BOTTOM
```

The Foundation demo uses the Variable feature extensively to simplify hand page editing. Basically, a Variable is a define that permits the substitution of text for a simple __VARIABLE__ string in a page. For example, in the test page above, the variables LEFTONLY_TOP and LEFTONLY_BOTTOM correspond to region templates that provide a logobar, menubar, leftside menu, and copyright footer. Content, consisting of HTML and ITL, is placed within the BEGIN and END CONTENT comments. The following illustration shows how this looks on the page:





In this diagram, LEFTONLY_TOP contributes the LEFTSIDE, LOGOBAR, and MENUBAR sections, while LEFTONLY_BOTTOM contributes the COPYRIGHT section.

The following subsections provide an inventory of where each of the region templates, included with the Foundation demo, are used in the pages and template definition files that make up the catalog.

4.5.1. LEFTONLY_TOP

The LEFTONLY_TOP template region is used in the following template pages:

```
pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/canceled.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/login.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/returns.html
pages/saved_carts.html
pages/ship_addresses.html
pages/ship_addresses_added.html
pages/ship_addresses_removed.html
pages/stock-alert-added.html
pages/stock-alert.html
```

The LEFTONLY_TOP template region is used in the following templates:

```
templates/foundation/cart
templates/foundation/leftonly
templates/foundation/simple
```

4.5.1.1. Region Template Walkthrough -- LEFTONLY_TOP

```
1 <!-- BEGIN LEFTONLY_TOP -->
2 [if scratch members_only]
3   [set members_only][set]
4   [if !session logged_in]
5     [set mv_successpage]@@MV_PAGE@[set]
6     [bounce page=login]
7   [/if]
8 [/if]
```

Foundation Store

```
9
10 <html>
11 <head>
12   <title>[scratch page_title]</title>
13   __THEME_CSS__
14 </head>
15
16 <body marginheight="0" marginwidth="0">
17
18 <!-- top left and right logo -->
19 <table width="100%" border="0" cellspacing="0" cellpadding="0">
20 <tr>
21   <td align="left" valign="middle" class="maincontent">
22     &nbsp;  
23   </td>
24   <td align="right" valign="middle" class="maincontent">
25     &nbsp;  
26   </td>
27 </tr>
28 </table>
29
30 <!-- menu bar along the top -->
31 <table width="100%" border="0" cellspacing="0" cellpadding="0">
32 <tr>
33   <td width="100%" class="menubar">
34     <a href="[area index]"></a>
35     
36     <a href="[area login]">
37       [if session logged_in]
38         </a>
39       [else]
40         </a>
41       [/else]
42     [/if]
43     
44     <a href="[area ord/basket]">
45     
46     <a href="[area ord/checkout]">
48     <a href="[area customerservice]">
50     <a href="[area aboutus]"></a>
51   </td>
52 </tr>
53 </table>
54
55 <!-- left category column, main content column, and right special column -->
56 <table width="100%" border="0" cellspacing="0" cellpadding="0">
57 <tr>
58   <td width="20%" valign="top" align="left" class="categorybar">
59     <!--Left Sidebar-->
60     <table width="100%" border="0" cellspacing="0" cellpadding="0">
61       [include file="templates/components/[control component none]"][control]
62       [include file="templates/components/[control component none]"][control]
63       [include file="templates/components/[control component none]"][control]
64     </table>
65   </td>
66   <td width="80%" valign="top" align="center" class="maincontent">
67     [include file="templates/components/[control component none]"][control]
68
```

4.5.2. LEFTONLY_BOTTOM

The LEFTONLY_BOTTOM template region is used in the following template pages:

```
pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/canceled.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/login.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/returns.html
pages/saved_carts.html
pages/ship_addresses.html
pages/ship_addresses_added.html
pages/ship_addresses_removed.html
pages/stock-alert-added.html
pages/stock-alert.html
```

The LEFTONLY_BOTTOM template region is used in the following templates:

```
templates/foundation/cart
templates/foundation/leftonly
templates/foundation/simple
```

4.5.3. LEFTRIGHT_TOP

The LEFTRIGHT_TOP template region is used in the following template pages:

```
pages/browse.html
pages/index.html
pages/results.html
pages/results_big.html
pages/swap_results.html
```

The LEFTRIGHT_TOP template region is used in the following templates:

```
templates/foundation/leftright
```

4.5.4. LEFTRIGHT_BOTTOM

The LEFTRIGHT_BOTTOM template region is used in the following template pages:

```
pages/browse.html
```

```
pages/index.html
pages/results.html
pages/results_big.html
pages/swap_results.html
```

The LEFTRIGHT_BOTTOM template region is used in the following templates:

```
templates/foundation/leftright
```

4.5.5. NOLEFT_BOTTOM

The NOLEFT_BOTTOM template region is used in the following template pages:

```
pages/ord/checkout.html
pages/splash.html
```

The NOLEFT_BOTTOM template region is used in the following templates:

```
templates/foundation/fullwidth
```

4.5.6. NOLEFT_TOP

The NOLEFT_TOP template region is used in the following template pages:

```
pages/ord/checkout.html
pages/splash.html
```

The NOLEFT_TOP template region is used in the following templates:

```
templates/foundation/fullwidth
```

4.6. Template Page List

/home/ic/catalogs/ft/pages/:

```
aboutus.html
account.html
browse.html
canceled.html
change_password.html
contact.html
customerservice.html
deliver.html
flypage.html
help.html
index.html
login.html
logout.html
lost_password.html
modular_modify.html
new_account.html
privacypolicy.html
process_return.html
quantity.html
results_big.html
```

```
results_either.html
results.html
returns.html
saved_carts.html
ship_addresses_added.html
ship_addresses.html
ship_addresses_removed.html
splash.html
stock-alert-added.html
stock-alert.html
swap_results.html
```

/home/ic/catalogs/ft/pages/admin/report_def:

```
Order%20Status.html
Products%20to%20edit.html
```

/home/ic/catalogs/ft/pages/admin/reports:

```
Order%20Status.html
Products%20to%20edit.html
```

/home/ic/catalogs/ft/pages/affiliate:

```
index.html
login.html
```

/home/ic/catalogs/ft/pages/ord:

```
basket.html
checkout.html
```

/home/ic/catalogs/ft/pages/query:

```
check_orders.html
get_password.html
order_detail.html
order_return.html
```

4.7. Special Page List

/home/ic/catalogs/ft/special_pages/:

```
badsearch.html
canceled.html
cc_not_valid.html
confirmation.html
failed.html
interact.html
missing.html
needfield.html
nomatch.html
noproduct.html
notfound.html
order_security.html
reconfig.html
sec_faq.html
```

```
security.html
violation.html
```

4.8. Components

- Added new [control] and [control–set] tags to set series of Scratch– like option areas. Used for components in UI content editing.

Interchange components are portions of HTML and ITL that are included in pages within a catalog depending on options set in the Administration Tool. The default component set includes the following:

```
affiliate_receptor
best_horizontal
best_vertical
cart
cart_display
cart_tiny
category_vertical
cross_horizontal
cross_vertical
modular_buy
modular_update
none
promo
promo_horizontal
promo_vertical
random
random_horizontal
random_vertical
saved_carts_list_small
search_box_small
upsell
upsell_horizontal
upsell_vertical
```

/home/ic/catalogs/ft/templates/components:

4.8.1. affiliate_receptor

Not used in Foundation demo

4.8.2. best_horizontal

The best_horizontal component is used in the following templates:

```
templates/foundation/cart
templates/foundation/leftonly
templates/foundation/leftright
```

Not used in Foundation demo pages

4.8.3. best_vertical

The best_vertical component is used in the following template:

```
templates/foundation/leftright
```

Not used in Foundation demo pages

4.8.4. cart

The cart component is used in the following page:

```
pages/ord/basket.html
```

4.8.5. cart_display

The cart_display component creates a small shopping cart that is displayed on the search results page (pages/results.html). It is displayed after an item in a list of results from a search is added to the shopping cart. cart_display is called in results.html by the following include statement:

```
[include file="templates/components/cart_display"]
```

The cart_display component is used in the following pages:

```
pages/results.html
```

4.8.5.1. Component Walkthrough -- cart_display

The remainder of this section is best read in conjunction with the file CATROOT/templates/components/cart_display in a text editor.

Lines 1–6: Component Specification

```
1 [comment]
2 ui_component: cart_display
3 ui_component_group: info
4 ui_component_label: Smaller cart for display in content area
5
6 [/comment]
7
```

These lines control what is shown in the Edit page screen of the admin interface.

```
8 <!-- BEGIN COMPONENT [control component cart_display] -->
```

Line 8 is an HTML comment noting the start of the code for the component. (Note that this can serve as a useful debugging tool to help you locate the component in the resulting HTML generated by Interchange when you view the source of a page loaded in the browser.)

```
9 [if items]
```

Line 9 checks to see if there are items in the shopping basket. If there are, the remaining code up to the closing [/if] tag on line 64 is executed. If not, Interchanges continues executing the remaining code in results.html (the file that calls the cart_display component).

```
10 <center>
11 <table width="95%" border="0" cellspacing="0" cellpadding="0">
```

Foundation Store

```
12     <TR class="contentbar2" VALIGN=TOP>
13         <td align=center class="contentbar2">Action</td>
14         <td class="contentbar2">
15             SKU
16         </td>
17         <td class="contentbar2">
18             Description
19         </td>
20         <td class="contentbar2">
21             Quantity
22         </td>
23         <td class="contentbar2">
24             Price
25         </td>
26         <td class="contentbar2">
27             Extension
28         </td>
29     </TR>
```

Line 10 centers the table started in line 11. Lines 12–29 create a header row in the shopping cart consisting of the header titles Action, SKU, Description, Quantity, Price, and Extension.

```
30     <TBODY>
31     [item-list]
32
```

Line 30 defines the remainder of the table as a section while the [item-list] tag on line 31 tells Interchange to execute the code up to the closing tag ([/item-list] on line 59 for each item the customer has ordered so far.

```
33     <tr class="[item-alternate 2]maincontent[else]contentbar1[/else][/item-alternate]">
34         <td align=center valign=top>
35             [page ord/basket]edit</A>
36         </TD>
37         <td valign=top>[item-code]</TD>
38         <td valign=top>[page [item-code]][item-description]</A>
39         </TD>
40
```

Line 33 begins the next row in the table. The [item-alternate] tag provided as the value of the class attribute tells Interchange to alternate between displaying the rows according to the "maincontent" and "contentbar1" styles (gray and white, respectively).

Lines 34–36 create a link to the shopping cart (basket.html) where the customer can remove or change the quantity of the item ordered.

Line 37 displays the SKU of the item. Lines 38 and 39 provide a link to the product display page (flypage.html) for the item. The [item-description] tag providing the content of the [page] tag enables the item's name to be displayed as the link to the product display page.

```
41     [if-item-modifier gift_cert]
42         <TD ALIGN=CENTER><small>Amount of gift:</small></TD>
43         <TD ALIGN=CENTER>[item-quantity]</TD>
44         <TD ALIGN=right>
45             [item-subtotal]
46         </TD>
47     [else]
48         <TD ALIGN=CENTER>[item-quantity]</TD>
49         <TD ALIGN=right>
```

```

50         [item-price]
51     </TD>
52     <TD ALIGN=right>
53         [item-subtotal]
54     </TD>
55 [/else]
56 [/if-item-modifier]
57 </TR>
58

```

Line 41 checks whether the item is a gift certificate. If it is it displays "Amount of gift:" and the [item-quantity] (number of gift certificates, in this case) under the headings "Quantity" and "Price", respectively. Otherwise, lines 48 through 50 display the quantity and price of the item ordered. Lines 45 or 53 (depending on whether the item is a gift certificate) display the item subtotal (quantity multiplied by price) for the item under the heading "Extension".

```

59 [/item-list]
60 </TBODY>
61 </table>
62 </FORM>
63 </center>
64 [/if]
65
66 <!-- END COMPONENT [control component cart_display] -->

```

Lines 59 through 64 close out the tags for the component, and line 66 indicates the end of the component code.

4.8.6. cart_tiny

The cart_tiny component is used in the following pages:

```

pages/account.html
pages/browse.html
pages/canceled.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/index.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/saved_carts.html
pages/ship_addresses.html

```

4.8.7. category_horizontal

Not used in Foundation demo pages or templates.

4.8.8. category_vertical

The `category_vertical` component provides a listing of all products in the catalog, organized by `prod_group` (e.g., Hand Tools, Ladders). `category_vertical` is usually displayed in the LEFTSIDE section of the page, under the `search_box_small` component.

The `category_vertical` component is used in the following pages:

```
pages/aboutus.html
pages/account.html
pages/affiliate/index.html
pages/affiliate/login.html
pages/browse.html
pages/canceled.html
pages/contact.html
pages/customerservice.html
pages/flypage.html
pages/help.html
pages/index.html
pages/login.html
pages/logout.html
pages/modular_modify.html
pages/new_account.html
pages/ord/basket.html
pages/privacypolicy.html
pages/process_return.html
pages/quantity.html
pages/query/check_orders.html
pages/query/order_detail.html
pages/query/order_return.html
pages/results.html
pages/results_big.html
pages/returns.html
pages/saved_carts.html
pages/ship_addresses.html
pages/stock-alert-added.html
pages/stock-alert.html
pages/swap_results.html
```

4.8.8.1. Component Walkthrough -- category_vertical

The remainder of this section is best read while viewing the file `CATROOT/templates/components/cart_display` in a text editor.

Lines 1–6: Component Specification

```
1 [comment]
2 ui_component: category_vertical
3 ui_component_group: Navigation
4 ui_component_label: Vertical category list
5
6 page_class:
7     label: Page class
8     widget: select
9     lookup: which_page
10    db: area
11    help: Defines which sets of items should be displayed
12    advanced: 1
```

Foundation Store

```
13
14 set_selector:
15     label: Page area selector
16     widget: select
17     db: area
18     lookup: sel
19     help: Defines which sets of items should be displayed
20     advanced: 1
21 [/comment]
22
```

These lines control what is shown in the Edit page screen of the Administration Tool.

```
23 <tr><td align="center" class="categorybar">
24     <br>
25     <table>
26
27 <!-- BEGIN COMPONENT [control component category_vertical] -->
```

Lines 23–25 set up the row and table within that row that will hold the vertical category list. Line 27 identifies the start of the code for the list.

```
28 [loop
29     prefix=box
30     search="
31         fi=area
32         st=db
33         tf=sort
34         ac=0
35         ac=0
36         co=yes
37
38         sf=sel
39         op=eq
40         se=[control set_selector left]
41
42         sf=which_page
43         op=rm
44         se=[control page_class all|@@MV_PAGE@@]
45 "]"
46
```

Lines 28–45 build a list of product categories obtained through a search of the area table.

```
47 <tr>
48     <td valign="top" class="categorybar">
49         <b>[box-exec bar_link]area[/box-exec]</b>
50     </td>
51 </tr>
52 <tr>
53     <td valign="top" class="categorybar">
54
55 [set found_cat][[/set]
56 [loop prefix=cat
57     search="
58         fi=cat
59         st=db
60         tf=sort
61         tf=name
62         rf=code,name
```


4.8.12. modular_update

The modular_update component is used in the following pages:

```
pages/modular_modify.html
```

4.8.13. promo

The promo component is used in the following pages:

```
pages/contact.html  
pages/results_big.html
```

4.8.14. promo_horizontal

The promo_horizontal component is used in the following pages:

```
pages/aboutus.html  
pages/canceled.html
```

The promo_horizontal component is used in the following templates:

```
templates/foundation/cart  
templates/foundation/leftonly  
templates/foundation/leftright
```

4.8.15. promo_vertical

Not used in Foundation demo pages.

The promo_horizontal component is used in the following templates:

```
templates/foundation/leftright
```

4.8.16. random

The random component is used in the following pages:

```
pages/browse.html  
pages/index.html  
pages/ord/basket.html  
pages/privacypolicy.html  
pages/process_return.html  
pages/results.html  
pages/swap_results.html
```

4.8.17. random_horizontal

Not used in Foundation demo pages.

The random_horizontal component is used in the following templates:

```
templates/foundation/cart  
templates/foundation/leftonly  
templates/foundation/leftright
```

4.8.18. random_vertical

Not used in Foundation demo pages.

The random_vertical component is used in the following templates:

```
templates/foundation/leftright
```

4.8.19. saved_carts_list_small

The saved_carts_list_small component is used in the following pages:

```
pages/ord/basket.html
```

4.8.20. search_box_small

The search_box_small component is used in the following pages:

```
pages/aboutus.html  
pages/account.html  
pages/affiliate/index.html  
pages/affiliate/login.html  
pages/browse.html  
pages/canceled.html  
pages/contact.html  
pages/customerservice.html  
pages/flypage.html  
pages/help.html  
pages/index.html  
pages/login.html  
pages/logout.html  
pages/modular_modify.html  
pages/new_account.html  
pages/ord/basket.html  
pages/privacypolicy.html  
pages/process_return.html  
pages/quantity.html  
pages/query/check_orders.html  
pages/query/order_detail.html  
pages/query/order_return.html  
pages/results.html  
pages/results_big.html  
pages/returns.html  
pages/saved_carts.html  
pages/ship_addresses.html  
pages/stock-alert-added.html  
pages/stock-alert.html  
pages/swap_results.html
```

The search_box_small component is used in the following templates:

```
templates/regions/LEFTONLY_TOP  
templates/regions/LEFTRIGHT_TOP
```

4.8.21. upsell

Not used in Foundation demo pages.

4.8.22. upsell_horizontal

The upsell_horizontal component is used in the following pages:

```
pages/flypage.html
```

The upsell_horizontal component is used in the following templates:

```
templates/foundation/cart  
templates/foundation/leftonly  
templates/foundation/leftright
```

4.8.23. upsell_vertical

Not used in Foundation demo pages.

The upsell_vertical component is used in the following templates:

```
templates/foundation/leftright
```


5. The Database Tables

Interchange catalogs are centralized around the database. You can alter any of the standard databases, add new databases, or remove unneeded databases

The foundation catalog includes the following tables, organized here by content:

- Your site content data
 - area.txt
 - cat.txt
 - downloadable.txt
 - merchandising.txt
 - options.txt
 - pricing.txt
 - products.txt
 - products.txt.category
- Customer data
 - access.asc
 - gift_certs.txt
 - userdb.txt
- Transaction–related data
 - inventory.txt
 - orderline.txt
 - order_returns.txt
 - transactions.txt
- Third–party relationship data
 - affiliate.txt
 - banner.txt
- Site administrative data
 - component.txt
 - files.txt
 - ichelp.txt
 - icmenu.txt
 - locale.txt
 - mv_metadata
 - route.txt
 - shipping.asc
 - variable.txt
- Ancillary data
 - 2ndDayAir.csv
 - 450.csv
 - country.txt
 - Ground.csv
 - NextDayAir.csv
 - salestax.asc
 - state.txt

The following dictionary lists and describes each table used in the Foundation demo.

5.1. 2ndDayAir.csv

Shipping table from UPS (<http://www.ups.com/using/services/rave/rate/>). This and all shipping tables should be updated periodically.

5.2. 450.csv

Shipping table from UPS for 450xx Zip Code origin. You will probably need to get your own from the UPS site (<http://www.ups.com/using/services/rave/rate/>) and clip the headers.

5.3. Ground.csv

Shipping table from UPS (<http://www.ups.com/using/services/rave/rate/>).

5.4. NextDayAir.csv

Shipping table from UPS (<http://www.ups.com/using/services/rave/rate/>).

5.5. access.asc

Administrative access table. This table is used by the Administration Tool. For more description on these fields, see the Red Hat Interchange Administration Tool guide.

Fields

<i>Field</i>	<i>Description</i>
username	Login name or group name (group names begin with ':')
password	Hashed password
name	Administrator's name
last_login	Last login time
super	Set to 1 if superuser
yes_tables	Tables the user may edit
no_tables	Tables the user may not edit
upload	No Description
acl	No Description
export	No Description
edit	No Description
pages	No Description
files	No Description
config	No Description
reconfig	No Description
groups	Administrator's group memberships

meta	No Description
no_functions	Explicitly disallowed functions
yes_functions	Allowed functions with permission flags
table_control	No Description
personal_css	Administrator's personal CSS (for admin screen presentation)

5.5.1. username

Example Data

```
:ausers
:busers
BigUser
goody
ic
```

The login name for an administrator or an administration group. Group names are prefixed with a colon (:).

5.5.2. password

Example Data

```
Ksjs65bMNLjPQ
```

Hashed password.

5.5.3. name

Example Data

```
Interchange Site Administrator
Interchange Site Associates
Business Users
2nd Shift
Mr. Jones
Inbound Sales
```

Descriptive name for the administrator or administration group.

5.5.4. last_login

Example Data

```
989424489
```

Last login time (in unix time() format).

5.5.5. super

Boolean value. If true (1), the administrator has Interchange Site Administrator privilege.

5.5.6. yes_tables

Example Data

```
affiliate=vex component=v gift_certs=v inventory=vx ...  
NONE
```

Tables this administrator or administration group can access. This is a space-delimited list of 'table_name=permission_flags' entries.

5.5.7. no_tables

Example Data

```
access mv_metadata variable
```

Tables this administrator or administration group can not use. This is a space-delimited list of tables names.

5.5.8. upload

No Description

5.5.9. acl

No Description

5.5.10. export

No Description

5.5.11. edit

No Description

5.5.12. pages

No Description

5.5.13. files

No Description

5.5.14. config

No Description

Example Data

Allowed Values

5.5.15. reconfig

No Description

5.5.16. groups

Example Data

```
  ausers
  busers
```

Allowed Values

Groups the site user belongs to. You can set permissions for groups.

5.5.17. meta

No Description

5.5.18. no_functions

Example Data

```
  orderstats trafficstats
```

Space-delimited list of functions explicitly not allowed for the site user.

5.5.19. yes_functions

Example Data

```
  item=lvecd itemtype=lvc order=lvca orderstats trafficstats ...
  NONE
```

Functions the site user can perform. This is a space-delimited list of functions, with permission flags if appropriate.

Usage examples

- dist/lib/UI/pages/admin/access_permissions.html

5.5.20. table_control

No Description

Usage examples

- dist/lib/UI/Primitive.pm
- dist/lib/UI/pages/admin/special/key_violation.html
- dist/lib/UI/usertag/if_mm

5.5.21. personal_css

Used in the Administration Tool screens to make personal changes to the page presentation. This is done by creating your own personal CSS (cascading style sheet).

Usage examples

- `dist/lib/UI/pages/admin/preferences.html`

5.6. affiliate

`cat_root/products/affiliate.txt`

This table contains data related to your affiliate programs. See also the [affiliate_receptor](#) component.

Fields

<i>Field</i>	<i>Description</i>
affiliate	Affiliate ID
name	Name of affiliate organization
campaigns	Campaigns this affiliate participates in
coupon_amount	Discount for customers from affiliate participating in coupon campaign
join_date	When the affiliate signed with you
url	Your default URL to use for customers coming from the affiliate site (not the affiliate's home page)
timeout	Timeout in seconds after which purchases are no longer credited to the affiliate
active	Boolean, set to 1 for active affiliates
password	Affiliate login password
image	Affiliate's logo

5.6.1. affiliate

Example Data

```
consolidated
hardhat
```

This field contains the unique Affiliate ID.

5.6.2. name

Example Data

```
Consolidated Diversified
Hardhat Construction
```

This is the descriptive name of the affiliate.

5.6.3. campaigns

Example Data

```
coupon
```

This field lists the campaigns that the affiliate participates and enables campaign features and tracks traffic from advertising campaigns. The foundation catalog implements a coupon campaign in the `affiliate_receptor` component. If you want to add campaigns, you will also need to develop the appropriate logic within the `affiliate_receptor` component and pages that use it.

5.6.4. coupon_amount

Example Data

```
5
```

This is the discount offered customers from the affiliate participating in the coupon campaign.

Note — This is implemented in the `affiliate_receptor` component as a flat discount amount. If you wanted a percentage discount instead, you would modify the `[discount]` tag in `catalog_root/templates/components/affiliate_receptor` (see the `[discount]` tag for more detail).

5.6.5. join_date

Example Data

```
20000827
20000910
```

This is the date when the affiliate signed with you.

5.6.6. url

Example Data

```
http://demo.akopia.com/~hardhat
http://www.minivend.com/consolidated/
```

The value in this field is used to direct visitors coming from the Affiliate to your home page or a page you have designed specifically for visitors from that Affiliate's site. Note that this should *not* be the URL of the Affiliate's home site.

5.6.7. timeout

Example Data

```
0
3600
```

The value in this field is used to specify the amount of time a customer has to place an order to still give the Affiliate credit for it. If the customer goes over this amount of time, the Affiliate doesn't get credit for the customer visit. The timeout delay is measured in seconds, with the value of 0 (zero) disabling it. It is recommended that you use a value in the thousands to make sure the customer has enough time to shop.

5.6.8. active

This is a boolean value indicating whether the affiliate is active.

5.6.9. password

Example Data

akopia

Password for affiliate login (see `catalog_root/pages/affiliate/login.html`). Note that the password is stored in plaintext by default.

5.6.10. image

Example Data

`http://demo.akopia.com/~hardhat/images/logo.gif`
`http://www.minivend.com/consolidated/conslogo.gif`

Affiliate's logo image.

5.7. area

`cat_root/products/area.txt`

This table is used to implement dynamic navigation bars. For example, it is used in the `category_horizontal` and `category_vertical` components. Note the similarity to the `cat` table, since both `area` and `cat` tables supply data for building links to results pages.

When building entries in a navigation bar, it is the `bar_link` subroutine in the `/dist/catalog_before.cfg` configuration file that actually reads and processes the values from the table.

See also the following catalog and administrative templates:

- `cat_root/templates/components/category_horizontal`
- `cat_root/templates/components/category_vertical`
- `dist/lib/UI/pages/admin/layout.html`
- `dist/lib/UI/pages/admin/layout_auto.html`
- `dist/lib/UI/pages/admin/wizard/do_launch.html`
- `dist/lib/UI/pages/admin/wizard/do_save.html`

Fields

<i>Field</i>	<i>Description</i>
--------------	--------------------

code	Unique key
sel	Space-delimited list of navigation bars to contain the entry
name	Display label
which_page	Page class in which the navigation bar may appear
sort	Sorting prefix for entry (preempts standard alphanumeric sort)
display_type	How to label links in the navbar (name, icon, url or image)
image	Image URL (if appropriate)
image_prop	HTML attributes for output tag (if appropriate)
banner_image	Image name for use in target page
banner_text	Text for use in target page
link_type	Type of links in the navbar (external, internal, simple, complex)
url	Target for internal or external link_type
tab	Database table file to use with 'simple' link_type
page	Results page to use with 'simple' link_type
search	Search spec used with 'complex' link_type
selector	The selector used to scan the <code>products</code> table for products in the category
link_template	Overrides template used for building navbar links.

5.7.1. code

Example Data

```
1
2
3
```

Unique key.

5.7.2. sel

Example Data

```
left
```

Space-delimited list of navigation bars that should contain the entry. Note that comma or null should also work as a delimiter.

5.7.3. name

Example Data

```
Hand Tools
Hardware
Ladders
Measuring Tools
Painting Supplies
Safety Equipment
```

Specials
Tool Storage

Label to display.

5.7.4. which_page

Example Data

all

Page class in which the navigation bar may appear.

5.7.5. sort

Example Data

00
03
04
05
06

Lexographic (alphanumeric) sorting prefix. Note use of '03' rather than '3', which would sort after '13'. This controls the order of the categories in your navigation bar.

If this is not set, your navbar entries will sort in alphabetical order.

5.7.6. display_type

Example Data

name
icon
url
image

What to use for the labels in the navigation bar (for example, name, icon, url or image). The navigation bars in the foundation catalog are set up with 'name' display_type.

<i>display type</i>	<i>Link shown as</i>
name	Displays name only
icon	Displays name and specified image
image	Displays image only
url	Displays link

5.7.7. image

Image URL for image or icon **display_type**.

5.7.8. image_prop

For image or icon **display_type**, this contains the HTML attributes for the HTML that will appear in the navbar, for example:

```
name
```

5.7.9. banner_image

Example Data

```
promo_image.gif
```

This field is not related to banner ads. It is useful if you want to pass to your results page an image that is specific for the navbar entry (perhaps to display a banner above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' **link_type**), then this will add 'va=banner_image=**banner_image**' to the resulting search specification. This puts the contents of **banner_image** into the Values hash of your search results page. You can access it with [[value banner_image](#)] (see the [value](#) tag). You will have to modify the standard results page (or set up and specify your own) in order to display the image.

```
<IMG src="[value banner_image]" alt="[value banner_text]">
```

The foundation catalog does not implement banner_image in the preconfigured navigation bars.

5.7.10. banner_text

Example Data

```
This Is A Title For Hand Tools
```

This field is not related to banner ads. It is useful if you want to pass to your results page some text that is specific for the navbar entry (perhaps to display a title above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' **link_type**), then this will add 'va=banner_text=**banner_text**' to the resulting search specification. This puts the contents of **banner_text** into the Values hash of your search results page. You can access it with [[value banner_text](#)] (see the [value](#) tag). You will have to modify the standard results page (or set up and specify your own) in order to display the text.

The foundation catalog does not implement banner_text in the preconfigured navigation bars.

5.7.11. link_type

Example Data

```
none
external
internal
simple
```

complex

Link type to create in the navigation bar.

<i>Link type</i>	<i>Description</i>
none	No link
external	External link. The HTML specified in url will go directly into the navigation bar.
internal	Internal link. This will be highlit if it is the current page. If you specify both a page and a form for the link, the url field should contain " <i>page form</i> ". See the Search Engine documentation for more detail on search forms.
simple	Allows you to specify an Interchange search with a few values. See the bar_link subroutine in the <code>/dist/catalog_before.cfg</code> configuration file for more detail.
complex	Allows you to fully specify an Interchange search specification. See the bar_link subroutine in the <code>/dist/catalog_before.cfg</code> configuration file for more detail if you need to use these.

5.7.12. url

Target URL (external link or internal page/search specification). See **link_type** . The foundation catalog navigation bars are not set up with link types that use the **url** field.

5.7.13. tab

Database table file to use with 'simple' **link_type** (searchspec fi=**tab**). The foundation catalog navigation bars are not set up with link types that use the **tab** field.

5.7.14. page

Results page to use with 'simple' **link_type** (searchspec sp=**page**). The foundation catalog navigation bars are not set up with link types that use the **page** field.

5.7.15. search

Search spec used with 'complex' **link_type**. See the Search Engine documentation for more detail on search forms. The foundation catalog navigation bars are not set up with link types that use the **search** field.

5.7.16. selector

The selector that is used to scan the `products` table for products in the category. Used with 'simple' **link_type**. The foundation catalog navigation bars are not set up with link types that use the **selector** field.

5.7.17. link_template

Overrides the usual HTML link template for navbar entries. See the **bar_link** subroutine in the `/dist/catalog_before.cfg` configuration file if you need to modify link templates.

The foundation catalog navigation bars are not set up with link types that use the **link_template** field.

5.8. banner

cat_root/products/banner.txt

The banner ad table. The foundation catalog does not implement any banner ads with this table.

You do not need to use this table to display ads served by third parties (for example, doubleclick). Since most banner ads on the internet are served by third parties and are not managed by your catalog, you probably will not need to set up banners here unless you do your own advertising.

See [Banner/Ad rotation](#) in the template documentation for a detailed description of the columns and content of the banner table. Also, see the [banner](#) tag documentation.

Fields

<i>Field</i>	<i>Description</i>
code	Key for the item. If the banners are not weighted, this should be a category-specific code.
category	Category for set of weighted banners
weight	Display frequency weight for weighted banner
rotate	Boolean: parse banner field for banners to rotate if true (1)
banner	Banner name or list of banners to rotate

5.8.1. code

Example Data

```
MyBanner
MyBanner2
MyBanner3
default
```

See [Banner/Ad rotation](#).

5.8.2. category

Example Data

```
BannerCat1
```

See [Banner/Ad rotation](#).

5.8.3. weight

Example Data

```
1
2
7
```

See [Banner/Ad rotation](#).

5.8.4. rotate

Boolean value. If true (1), rotates banners listed in **banner**. See [Banner/Ad rotation](#).

5.8.5. banner

Example Data

```
Default banner 1{or}Default banner 2{or}Default banner 3
First MyBanner
Second MyBanner
Third MyBanner
```

See [Banner/Ad rotation](#).

5.9. cat

cat_root/products/cat.txt

This table contains properties of product categories. Notice the similarity to the **area** table, since both the **area** and **cat** tables supply data for building links to results pages.

Fields

<i>Field</i>	<i>Description</i>
code	Unique key
sel	Space-delimited list of foreign keys into area table
name	Category name
which_page	Page class in which the category may appear
sort	Sorting prefix for entry (preempts standard alphanumeric sort)
display_type	How to label the category links (name, icon, url or image)
image	Image URL (if appropriate)
image_prop	HTML attributes for output tag (if appropriate)
banner_image	Image name for use in target page
banner_text	Text for use in target page
link_type	Type of links in the navbar (external, internal, simple, complex)
url	Target for internal or external link_type
tab	Database table file to use with 'simple' link_type
page	Results page to use with 'simple' link_type
search	Search spec used with 'complex' link_type
selector	The selector used to scan the products table for products in the category
link_template	Overrides template used for building links

5.9.1. code

Example Data

```
1
4
5
```

Unique key.

5.9.2. sel

Example Data

```
6
8 9
9
```

Space-delimited list of foreign key(s) into **area** table. The category will appear in each navbar section (defined by a row in the **area** table) where the key from **cat.sel** matches the **area.code**.

For example, the foundation catalog (tools) places Gift Certificates in more than one category of the left navbar.

5.9.3. name

Example Data

```
Breathing Protection
Eye Protection
Gift Certificate
Picks & Hatchets
Pliers
Rulers
Sandpaper
Toolboxes
```

Category name for display.

5.9.4. which_page

The page class. When building links, you can select categories matching a page class. This means you could set up your catalog to show a different list of links on page 'foo.html' than on page 'bar.html'.

5.9.5. sort

Example Data

```
01
03
```

Lexographic (alphanumeric) sorting prefix. Notice use of '03' rather than '3', which would sort after '13'. You

can use this to control the order of the categories in a list of links.

5.9.6. display_type

Example Data

```
name
icon
url
image
```

What to use for the labels in the navigation bar (for example, name, icon, url or image). The links in the foundation catalog are set up with 'name' display_type.

<i>display type</i>	<i>Link shown as</i>
name	Displays name only
icon	Displays name and specified image
image	Displays image only
url	Displays link

5.9.7. image

Image URL for image or icon **display_type**.

5.9.8. image_prop

For image or icon **display_type**, this contains the HTML tag attributes for the links, for example:

```
name
```

5.9.9. banner_image

Example Data

```
promo_image.gif
```

This field is not related to banner ads. It is useful if you want to pass to your results page an image that is specific for the navbar entry (perhaps to display a banner above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' **link_type**), then this will add 'va=banner_image=**banner_image**' to the resulting search specification. This puts the contents of **banner_image** into the Values hash in your search results page. You can access it with [[value banner_image](#)] (see the [value](#) tag). You will have to modify the standard results page (or set up and specify your own) in order to display the image. For example, you might include the following in your results page:

```
<IMG src="[value banner_image]" alt="[value banner_text]">
```

5.9.10. banner_text

This field is not related to banner ads. It is useful if you want to pass to your results page some text that is specific for the navbar entry (perhaps to display a title above your results).

If you are using an Interchange search for your links (i.e., 'simple' or 'complex' **link_type**), then this will add 'va=banner_text=**banner_text**' to the resulting search specification. This puts the contents of **banner_text** into the Values hash in your search results page. You can access it with [[value](#) banner_text] (see the [value](#) tag). You will have to modify the standard results page (or set up and specify your own) in order to display the text.

5.9.11. link_type

Example Data

```
none
external
internal
simple
complex
```

The link type to create.

<i>Link type</i>	<i>Description</i>
none	No link
external	External link. The HTML specified in url will go directly into the link.
internal	Internal link. This will be highlit if it is the current page. If you specify both a page and a form for the link, the url field should contain " <i>page form</i> ". See the Search Engine documentation for more detail on search forms.
simple	Allows you to specify an Interchange search with a few values. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail.
complex	Allows you to fully specify an Interchange search specification. See the bar_link subroutine in the /dist/catalog_before.cfg configuration file for more detail if you need to use these.

5.9.12. url

Target URL (external link or internal page/search specification). See **link_types** above.

5.9.13. tab

Example Data

```
products
```

Database table file to use with 'simple' **link_type** (searchspec fi=**tab**).

5.9.14. page

Example Data

```
swap_results
```

Results page to use with 'simple' **link_type** (searchspec sp=**page**).

5.9.15. search

Example Data

```
fi=merchandising^Msf=featured^Mse=new
fi=merchandising^Msf=featured^Mse=special^Msu=yes
```

Search spec used with 'complex' **link_type**. See the Search Engine documentation for more detail on search forms.

Note: The '^M' delimiters in the sample data represents a carriage return character (Control-M, or hexadecimal 0x0d).

5.9.16. selector

Example Data

```
category=Breathing Protection
category=Eye Protection
category=Gift Certificate
category=Picks & Hatchets
category=Pliers
category=Rulers
category=Sandpaper
category=Toolboxes
```

The element that is used to scan the `products` table for products in the category. Used with 'simple' **link_type**.

5.9.17. link_template

Overrides the usual HTML link template for navbar entries. See the **bar_link** subroutine in the `/dist/catalog_before.cfg` configuration file if you need to modify link templates.

5.10. country

A list of countries used to build select boxes and shipping mode choices based on countries.

```
code
sorder
region
selector
shipmodes
name
```

5.11. downloadable

This table controls downloadable products. The Marketing Reports data set for the foundation catalog demonstrates downloadable products. List a product's **sku** in this table if you want to deliver it through a download. A customer can then download the file specified in the **dl_location** field after checkout.

For reference, see the implementation in the following files:

- catalog_root/pages/deliver.html
- catalog_root/etc/receipt.html
- catalog_root/pages/query/order_detail.html

Fields

<i>Field</i>	<i>Description</i>
sku	Unique key, matches product.sku
dl_location	Location of downloadable file
dl_type	MIME type of downloadable file

5.11.1. sku

Example Data

Example Data from the Marketing Reports data set:

```
00352as
22083da
49503cg
59330rt
59402fw
73358ee
83491vp
90773sh
```

This is the unique key for this table that is also the common key into the products table.

5.11.2. dl_location

Example Data from 'reports' catalog

```
download/00352as.pdf
download/22083da.pdf
download/49503cg.pdf
download/59330rt.pdf
download/59402fw.pdf
download/73358ee.pdf
download/83491vp.pdf
download/90773sh.pdf
```

File location of downloadable product.

5.11.3. dl_type

Example Data from 'reports' catalog

```
application/pdf
```

MIME type of downloadable content.

5.12. files.txt

A database where files (pages, etc.) can be kept instead of in the Unix filesystem.

5.13. gift_certs.txt

```
code
username
order_date
original_amount
redeemed_amount
available_amount
passcode
active
redeemed
update_date
```

5.14. inventory.txt

```
sku
  Quantity info
quantity
  Gets decremented after each sale.
stock_message
  The usual shipping time of the product.
  Out of stock message:
  In stock
  Ships in 3-5 days
  Ships in 4-6 weeks
  Special order
account
  Accounting info
  Sales account
cogs_account
```

5.15. locale.txt

```
code
en_US
de_DE
fr_FR
```

5.16. merchandising.txt

```
sku
featured
```

```
banner_text
banner_image
blurb_begin
blurb_end
    Closer (end text for feature display)
timed_promotion
start_date
    Start date
finish_date
upsell_to
    Cross-sell SKUs
cross_sell
cross_category
others_bought
times_ordered
```

5.17. mv_metadata

See the following sections in the icadvanced catalog for more information:

- [display tag and mv_metadata](#)
- [mv_metadata.asc](#)

5.18. options

This table contains data for implementing simple, matrix and modular options.

Simple options are options that a customer can combine arbitrarily, such as size and color. The selected options might affect price. See the [accessories](#) tag for more detail on option values for simple options.

Matrix options are preconfigured combinations of options. For example, if you sell titanium and carbon-fiber bike frames, but offer only certain combinations of frame material and color, your checkout page might include a select box with only the following entries:

- Silver Titanium: \$1672
- Black Titanium: \$1672
- Red Titanium: \$1674
- Black Carbon Fiber: \$1290
- Yellow Flame Carbon Fiber: \$1300

Note that there is no Yellow Flame Titanium offering, for example.

Modular options are like a structured bill of materials, where one product is a master item and other products are subitems for that master item. The subitems can also be master items to subitems at a lower level. In addition, subitems may be designated as 'phantom', which means that they are placeholders in the hierarchy of the structured bill of materials with their own subitems, but are not actual items themselves.

The foundation catalog with the computer data set uses modular options.

For more information, see the following pages and components in the foundation catalog:

- [cat_root/pages/flypage.html](#)

- cat_root/templates/components/modular_buy
- cat_root/templates/components/modular_update

Note: Subsequent foundation catalog releases may place simple, matrix, and modular option types in separate tables.

Fields

<i>Field</i>	<i>Description</i>
code	Unique ID for the product option
o_master	SKU of the master item for the option
sku	SKU for the option (foreign key into products table)
o_group	Product grouping code
o_sort	Sorting prefix for list display
phantom	Boolean --- Item is a phantom placeholder (as in structured bill of materials) with suboptions.
o_enable	Boolean --- enables suboptions for the option
o_matrix	Matrix-type option (preconfigured combinations of attributes)
o_modular	Modular-type option (master/subitem relationship like modular bill of materials)
o_default	Default selection for the option group or suboption for a phantom option
o_label	Short name for option display
o_value	Simple option values (in Interchange option format)
o_widget	The HTML widget to use for displaying the option group
o_footer	Not used in foundation catalog
o_header	Not used in foundation catalog
o_height	Height of widget (if applicable)
o_width	Width of widget
description	Option/Variant description (for description in display)
price	Price of this option/variant
wholesale	Dealer price of this option/variant
differential	Differential to add to the base item price when using a phantom bill of materials
weight	Weight difference with this option/variant (for shipping)
volume	Volume difference with this option/variant
mv_shipmode	No Description
o_exclude	Option groups to exclude (trumped by o_include). Modular only.
o_include	Option groups to include (trumps o_exclude). Modular only.

5.18.1. code

Example Data

1002
1003

1004
1005

Unique ID for the option.

5.18.2. o_master

Example Data

00010
999000
999001
999002

SKU of the master item for the option. The master item is one level up in the modular hierarchy, and must be one of the following:

- An item in the products table (matching **products.sk**u)
- Another option in the options table (matching **options.sk**u)
- A [phantom](#) item in the **options** table.

If an option has a master item, then a customer can not choose that option without having previously selected the master item.

The price for a master item is the sum of the master item's price and the price for each of the subitems. Because the subitems are recursively defined, the top-level item reflects the top level price plus the price of all selected options.

5.18.3. sku

Example Data

00010
999000
7000015
7000030

The sku for the item or option. This may not be unique for matrix options or if an option that belongs to multiple **o_masters** is listed for each master.

5.18.4. o_group

Example Data

A
B
C
I

Product group (scanned to see whether it applies to this product or not)

5.18.5. o_sort

Example Data

```
01
02
03
04
47
48
49
50
```

Sorting prefix for listing order of options.

5.18.6. phantom

Modular options only.

Boolean -- if true (1), then this is a phantom item acting as a placeholder for other items rather than an actual product. The item's **sku** will not match an entry in the products table, though the **o_master** will match either the **sku** of another phantom item in the **options** table or the **sku** of an item in the **products** table.

5.18.7. o_enable

Boolean -- Enables subitems for this item or option. Note that an option with **o_enable** false may itself still be a subitem for an option or item above it.

5.18.8. o_matrix

Boolean. Set true (1) for matrix-type options. See the options table in the tools data set for examples of matrix options. Matrix options that are part of a set have the same value for **options.sku**.

5.18.9. o_modular

Specifies a modular option. See main heading for description of modular options.

5.18.10. o_default

Example Data

```
1
11002
7000062
7000087
```

Selects the default option for a group.

5.18.11. o_label

Example Data

```
Add a second hard drive
Case Color
Case color
Case style
Include tapes
Red
```

This is the short name for option display.

5.18.12. o_value

Example Data

```
l=One 8GB tape,\r2=Two 8GB tapes,\r=None*
a=One 8GB tape,\rb=Two 8GB tapes,\r=None*
baby=Baby Tower,\rmid=Mid-tower,\rfull=Full Tower
baby=Baby tower,\rmid=Mid-tower,\rfull=Full tower
red=Passion Red,\rblue=Electric Blue,\rgreen=Sea Green,\rgrey=S...
red=Passion Red,\ryellow=Lemon Yellow,\rblue=Electric Blue,\rgr...
red=Rage Red,\ryellow=Honey Yellow
```

This is an Interchange value set for a simple option. It is typically a comma-delimited list of labels and values with '*' indicating the default value. See the [accessories](#) tag for more detail.

Note that the "\r" characters in the above example represent carriage returns in the actual data ("\\r" in perl, or Ctrl-M, or hexadecimal 0D), and the ... indicates a line too long to show.

5.18.13. o_widget

Example Data

```
select
```

This determines the HTML Widget type (*e.g.*, a select box). For example, the [\[options\]](#) tag uses this entry when building HTML widgets in a page. See also the [\[accessories\]](#) tag for available widgets.

5.18.14. o_footer

Example Data

Allowed Values

5.18.15. o_header

Example Data

Allowed Values

5.18.16. o_height

This allows you to set the height of the HTML widget, if appropriate.

5.18.17. o_width

This allows you to set the width of the HTML widget, if appropriate.

5.18.18. description

Example Data

```
ATX Mid Tower-Grey (3)5.25 (2)3.5 & (1)3.5 Hidden  
Enlight ATX Desktop Case (2)5.25 & (2)3.5  
Enlight ATX Tower Case (4) 5.25 & (2)3.5  
Micro ATX Tower - Honey Yellow  
Micro ATX Tower - Moody Blue  
Micro ATX Tower - Rage Red  
Micro ATX Tower - Smoky Grey  
Super Tower Case (6)5.25 & (3)3.5
```

Longer description to show when displaying the options.

5.18.19. price

Example Data

```
0.00  
10  
20  
29  
75
```

This sets the retail price of the option.

5.18.20. wholesale

Example Data

```
13  
40.00
```

This sets the dealer price of the option.

5.18.21. differential

Example Data

```
-209  
-40  
-79
```

The phantom bill of materials for an option group can have a differential, which is an amount to add to the base price of the master product to get to a new base price that accommodates the phantom bill of materials. Note that the differential can be negative.

For example, in the computer data set of the foundation catalog, SKU 00011 in the products table is an \$849.95 pre-configured Athlon 800MHz computer that includes a 17" monitor (in this case, SKU 7000087 in the products table).

The monitor by itself would otherwise have cost \$209. It is much more convenient if you can use the same option part number and price for each item. To do this, you need a phantom option (in this case, SKU 999105 in the options table only) with a differential of -209 and the available monitors as suboptions. When you include the phantom option in the bill of materials for the computer (SKU 00011), the \$-209 differential adjustment makes the price work out properly.

For instance, suppose that a \$499 computer is configured as follows:

500 MHz Athlon	--	\$499
32 MB SDRAM	--	ZERO
10 GB disk	--	ZERO
TOTAL	--	\$499

Suppose it costs \$90 to upgrade the base computer to 128M of RAM and \$150 for a 30 GB hard disk.

If you also sell an 128MB 800 MHz \$899 computer, and the customer upgrades to the 30 GB hard disk,

800 MHz Athlon	--	\$899
(memory differential)	--	\$-90
128 MB RAM	--	\$90
30 GB disk	--	\$150
TOTAL	--	\$1039

If you did not have the differential, you would need a different option part number for each item make the number come out right.

With the differential, you can use the same part number for 128MB RAM no matter what the base part is. The price is always \$90 — there is just a -90 differential when ordered with the 800MHz Athlon, making the effective price zero.

5.18.22. weight

Example Data

5

Shipping weight of the option. Interchange uses this to calculate shipping cost.

5.18.23. volume

Volume added by the option.

5.18.24. mv_shipmode

No Description

5.18.25. o_exclude

Modular options only.

Lists the option groups to exclude once the include has been done. Takes the form of a number of wildcard atoms.

5.18.26. o_include

Modular options only.

Lists the option groups to include with your item. Takes the form of a number of wildcard atoms.

5.19. order_returns.txt

```

code
order_number
session
username
rma_number
nitems
total
return_date
update_date

```

5.20. orderline.txt

Every line item that is actually ordered is detailed in this table. The order as a whole is one record in the transactions table.

See the page `query/check_orders.html` for how it can be used. See `etc/report` for how to add to it.

```

code
store_id
order_number
session
username
shipmode
sku
quantity
price
subtotal
shipping
taxable
mv_mi
mv_si
size
color
options

```

```

order_date
update_date
status
    pending = Pending
    shipped = Shipped
    backorder = Back ordered
    credit = Waiting for credit check
    canceled = Cancelled
parent
affiliate
campaign
description
mv_mp
    
```

5.21. pricing

This database works in conjunction with the `CommonAdjust` directive to allow quantity pricing for one product or for a group of products (sometimes known as mix-and-match). The fields `q2`, `q5`, `q10`, etc. are for the quantity levels; the `price_group` field selects the mix-and-match category for the product.

Fields

<i>Field</i>	<i>Description</i>
sku	Unique key, shared with products table
price_group	Mix-and-match category
q2	Retail, 2 or more
q5	Retail, 5 or more
q10	Retail, 10 or more
q25	Retail, 25 or more
q100	Retail, 100 or more
w2	Wholesale, 2 or more
w5	Wholesale, 5 or more
w10	Wholesale, 10 or more
w25	Wholesale, 25 or more
w100	Wholesale, 100 or more

5.21.1. sku

Example Data

```

os28004
os28006
os28057c
os28069
    
```

Unique key, matching the **sku** for an entry in products table.

5.21.2. price_group

Example Data

general

This field determines mix-and-match categories if you want to allow mix-and-match quantity pricing (i.e., where 5 of *these* plus 5 of *those* afford the **q10** price for both *these* and *those*).

5.21.3. q2

If set, this will be the price per item when the order quantity is 2 or greater.

5.21.4. q5

If set, this will be the price per item when the order quantity is 5 or greater.

5.21.5. q10

If set, this will be the price per item when the order quantity is 10 or greater.

5.21.6. q25

If set, this will be the price per item when the order quantity is 25 or greater.

5.21.7. q100

If set, this will be the price per item when the order quantity is 100 or greater.

5.21.8. w2

If set, this will be the dealer price per item when the order quantity is 2 or greater.

5.21.9. w5

If set, this will be the dealer price per item when the order quantity is 5 or greater.

5.21.10. w10

If set, this will be the dealer price per item when the order quantity is 10 or greater.

5.21.11. w25

If set, this will be the dealer price per item when the order quantity is 25 or greater.

5.21.12. w100

If set, this will be the dealer price per item when the order quantity is 100 or greater.

5.22. products

This is the main table for product data. See also '[The Product Database](#)' section in the database documentation.

The **sku** is also the master key in many of the related tables.

Fields

<i>Field</i>	<i>Description</i>
sku	Unique product ID
description	Short description for list display
title	Full title for book, CD, artwork, <i>etc.</i>
template_page	Not used in foundation catalog. No description.
comment	Longer description for item display (e.g., flypage.html)
thumb	Thumbnail image
image	Regular-sized image
price	Retail quantity one price
wholesale	Dealer minimum quantity price
prod_group	Product supercategory
category	Product category
nontaxable	Boolean. Set true (1) if nontaxable
weight	Weight in your units. Should match shipping table.
size	List of options used with accessories tag.
color	List of options used with accessories tag.
gift_cert	Boolean. Set true (1) if this is a gift certificate.
related	Deprecated in favor of merchandising.upsell_to
featured	Deprecated. Use merchandising table.
inactive	Boolean. Set true (1) to inactivate a product
url	Not Documented

5.22.1. sku

Example Data

```
gift_cert
os28004
os28006
os28057c
```

Unique identifier for the product. You should use only characters of the class A-Z a-z 0-9 _ - (i.e., matching the regular expression, '[-A-Za-z0-9_]+'). Although Interchange itself does not impose this

restriction, you may have problems with SQL databases, file systems, and URL encoding if you use other characters. For example, a slash (/) can interfere with URLs and filenames, a colon (:) can interfere with database representations (or file names on some operating systems), i<etc.>

5.22.2. description

Example Data

```
Brush Set
Disposable Brush Set
Ergo Roller
Gift Certificate
Painters Brush Set
Painters Ladder
Spackling Knife
Trim Brush
```

A short description for the product that is used for displaying in the shopping cart, receipt, and order report.

5.22.3. title

Example Data

```
Brush Set
Disposable Brush Set
Ergo Roller
Gift Certificate
Painters Brush Set
Painters Ladder
Spackling Knife
Trim Brush
```

This column is not used in the foundation catalog. Previously used in the Art store (simple) demo for a painting title. You probably want to use **description** instead.

You should modify the products and other tables to suit your catalog's requirements. You might use this field if you want to show titles for books, music, or other titled merchandise. If you do not use a title that is distinct from the short description, then you probably do not need this column in the table at all.

5.22.4. template_page

Not used in foundation catalog.

No Description.

5.22.5. comment

Example Data

```
A must have for all painters! This spackling knife is ergon...
Enjoy the perfect feel and swing of our line of hammers. Thi...
This set includes 2" and 3" trim brushes and our ergonomical...
This set of disposable foam brushes is ideal for any stainin...
```

This is the field for a long description of the product. If you are using an Interchange text/gdbm database, the field size is unlimited; if using another type of database, the length will be dependent on the field type selected. If you are using a SQL database, see the appropriate `cat_root/dbconf` subdirectory for a place to set `COLUMN_DEF` values. See also the database documentation, '[Importing from an ASCII File](#)', for details about defaults for columns that you do not define.

5.22.6. thumb

Example Data

```
gift_certificate.gif
os28004_b.gif
os28005_b.gif
os28006_b.gif
```

This is the filename for a small (thumbnail) image of the product.

5.22.7. image

Example Data

```
gift_certificate_large.gif
os28004.gif
os28005.gif
os28006.gif
```

This is the filename for a regular-sized image of the product, as it should appear in an HTML `` tag. You do not need to specify the path if the image files are in the usual Interchange image directory.

5.22.8. price

Example Data

```
1.00
12.99
14.99
9.99
```

The quantity-one price of the product. See the **wholesale** field and the [price](#) table for dealer and quantity pricing.

5.22.9. wholesale

Example Data

```
1
10
11
12
```

This is the minimum dealer price for the item. For quantity pricing, see the [price](#) table.

5.22.10. prod_group

Example Data

Hand Tools
Hardware
Ladders
Measuring Tools
Miscellaneous
Painting Supplies
Safety Equipment
Tool Storage

Product group (supercategory). This indicates the grouping of product categories, for example in the navigation bars created from the **area** table (note the match with the **name** data in the area table).

5.22.11. category

Example Data

Brushes
Gift Certificate
Hammers
Ladders
Nails
Paintbrushes
Putty Knives
Rollers

This is the category the product should appear in when you select a list. You can put a product in more than one category, but you may need to accommodate this in display and banner headings. Embedded perl is helpful for this.

5.22.12. nontaxable

Boolean value. If true (1), the sales tax calculation for an order will not include the cost of the product. See also the [salestax](#) tag.

5.22.13. weight

Example Data

1
2
3

This is a numeric value of the weight used for determining shipping costs (with UPS, for example). In the US, this is typically the weight in pounds in order to match the UPS, Fed Ex and other standard shipping tables.

5.22.14. size

Example Data

```
1", 2", 3"  
1', 1.5'  
1/4", 1/2", 3/4", 1", 2", 3"  
10oz, 15oz, 20 oz  
2"  
6'  
set  
standard, metric
```

This is where the old Construct Something demo store kept the 'size' options for a product. The foundation catalog now uses the **options** table instead to handle product options (also sometimes called product attributes).

The [accessories](#) tag can build HTML widgets from the comma-delimited list of product options. You can use a delimiter other than comma (if compatible with the table) as long as you also set the [delimiter](#) in the [accessories](#) tag.

You probably do not need this field if you use the options table (for example, if you are building from the foundation catalog).

5.22.15. color

Another product option column. No longer used in the foundation catalog. See **size** above for description.

5.22.16. gift_cert

Boolean value. If true (1), specifies that this product is a gift certificate. See also the [gift_certs](#) table.

5.22.17. related

Used for displaying "upsells," opportunities to purchase an additional item when this one is purchased. Contains a comma-separated list of SKUs to be offered.

The foundation catalog now instead uses the [upsell_to](#) field of the [merchandising](#) table for upselling.

5.22.18. featured

Deprecated in favor of the [merchandising](#) table.

5.22.19. inactive

If true (1), renders the product inactive (i.e., it will not appear in the catalog).

5.22.20. url

Not Documented

5.23. products.category.txt

The products.category.txt table (actually a link to products.txt.10) is an automatically generated index into the products table to speed category searches. See [Dictionary Indexing With INDEX](#) in the database documentation for details about auto-indexing of text databases.

5.24. route.txt

```
code
attach
continue
commit
commit_tables
counter
credit_card
cyber_mode
email
empty
encrypt
encrypt_program
errors_to
increment
inline_profile
individual_track
individual_track_ext
partial
pgp_cc_key
pgp_key
profile
receipt
reply
report
rollback
rollback_tables
supplant
track
```

5.25. saletax.asc

5.26. shipping.asc

Shipping methods table

5.27. state.txt

State/territory/county information

```
code
sorder
country
state
name
tax
postcode
shipmodes
tax_name
```

5.28. transactions.txt

Each individual customer order has an entry in this table. The line items are not entered here, but in the orderline table.

See the page `query/check_orders.html` for how it can be used. See `etc/report` for how to add to it.

```

code
store_id
order_number
session
username
shipmode
nitems
subtotal
shipping
handling
salestax
total_cost
fname
lname
    Last Name
company
address1
address2
    Address line 2
city
state
zip
country
phone_day
    Daytime Phone
phone_night
    Home Phone
fax
email
b_fname
b_lname
    Billing Last Name
b_company
b_address1
b_address2
    Billing Address Line 2
b_city
b_state
    Billing State
b_zip
    Billing Postcode
b_country
    Billing Country
b_phone
order_date
order_ymd
order_wday
payment_method
po_number
avs
order_id
update_date

```

```

status
affiliate
campaign
parent
archived
deleted
complete
comments

```

5.29. userdb.txt

The user database used for maintaining customer address information, account information, preferences, and more. See icdatabase for more information.

```

username
password
acl
mod_time
s_nickname
company
fname
lname
address1
address2
address3
city
state
zip
    Postcode
country
    Country
phone_day
mv_shipmode
b_nickname
b_fname
b_lname
b_address1
b_address2
b_address3
b_city
b_state
b_zip
b_country
b_phone
    Billing Phone
mv_credit_card_type
mv_credit_card_exp_month
mv_credit_card_exp_year
p_nickname
email
fax
phone_night
fax_order
    Payment method:
        (none) = Credit Card
        1 = Fax or Mail
        2 = Purchase order
        3 = COD
address_book
accounts

```

```
preferences
carts
owner
file_acl
db_acl
order_numbers
email_copy
mail_list
    Mailing lists the customer has joined:
        offer = Special offers
        newsletter = Newsletter
        alert = Alerts and Recalls
        upgrade = Upgrades
project_id
account_id
order_dest
credit_limit
inactive
dealer
    Dealer:
        (none) = No
        1 = Yes
b_company
feedback
???
```

5.30. variable.txt

Configuration database

```
code
    Variable name
Variable
pref_group
    Preferences area
```


6. HTML Hypertext links

Normally, regular hypertext links are not used in Interchange pages. These kinds of links will not include the session ID. If the customer follows an external link back to the catalog, the list of products ordered so far will have been lost. The `area` tag is used to generate a hypertext link which includes a session ID.

Instead of:

```
<A HREF="/cgi-bin/mv/shirts">Shirts</A>
```

Use:

```
<A HREF="[area shirts]">Shirts</A>
```


7. Images

Inline images are placed in Interchange pages in the normal fashion with ``. But since Interchange pages are served by a CGI program, do not use relative links. The Foundation store defines an image directory with the `ImageDir` and `ImageDirSecure` directives that automatically adjusts the image path to a set base directory.

8. Browser Cookies

The Foundation store enables the `cookies` directive so that users with cookie-capable browsers will retain session context. Then, standard `HREF` and `Interchange` page links can be intermixed without the fear of losing the shopping basket. Cookie capability is also required to use search caching, page caching, and statically generated pages. If the user's browser does not support cookies, the cache will be ignored.

If planning to use more than one host name within the same domain for naming purposes (perhaps a secure server and non-secure server), set the domain with the `CookieDomain` directive. This must contain at least two periods (.) as per the cookie specification, and must be located in the same server as the domain.

9. Dependencies in administration

In general, it's a good idea to leave fields empty if you don't want to use them, instead of removing them from the database altogether. That way nothing in admin or demo will break.

